[Final] Past Exams - 2020 #1700

Jero Wang ADMIN
2 years ago in Exam – Final

485
VIEWS

You can find the past exams here: https://cs61c.org/sp23/resources/exams/. Please check the linked past Piazza/Ed Q&A PDFs first before asking here. Many of the questions are already answered in those!

When posting questions, please reference the semester, exam, and question in this format so it's easier for students and staff to search for similar questions:

Semester-Exam-Question Number

For example: **SP22-Final-Q1**, or **SU22-MT-Q3**

Summer 2020 midterm walkthrough

Fall 2020 midterm Bit Manipulations Walkthrough

Fall 2020 midterm 1 Slip Walkthrough

Melanie McKune 2y #1700aaf ✓ Resolved

SU20-Final-Q5b part iii

Why are those the circuits chosen?

Rosalie Fang ADMIN 2y #1700aca

Multiple components to this.

1. What address are we using? so looking at circuits where the right side (result) of the mux should be MemAddr, we check to see what we want to actually be our MemAddr, our circuit is usually implemented with ALU being the MemAddr, so we want a mux where if we don't have jas instruction everything stays the same. The second input to the mux should be the address we're writing to, or sp-4 so choose D.
2. What data are we using? here we're looking at all the circuits where right side of the mux is WriteData. our circuit usually implements rs2 being the writeData, but now we want to write PC+4 to it as per the problem description, so we choose G.

1

Anonymous Lapwing 2y #1700aac ✓ Resolved

SU20-Final-Q6

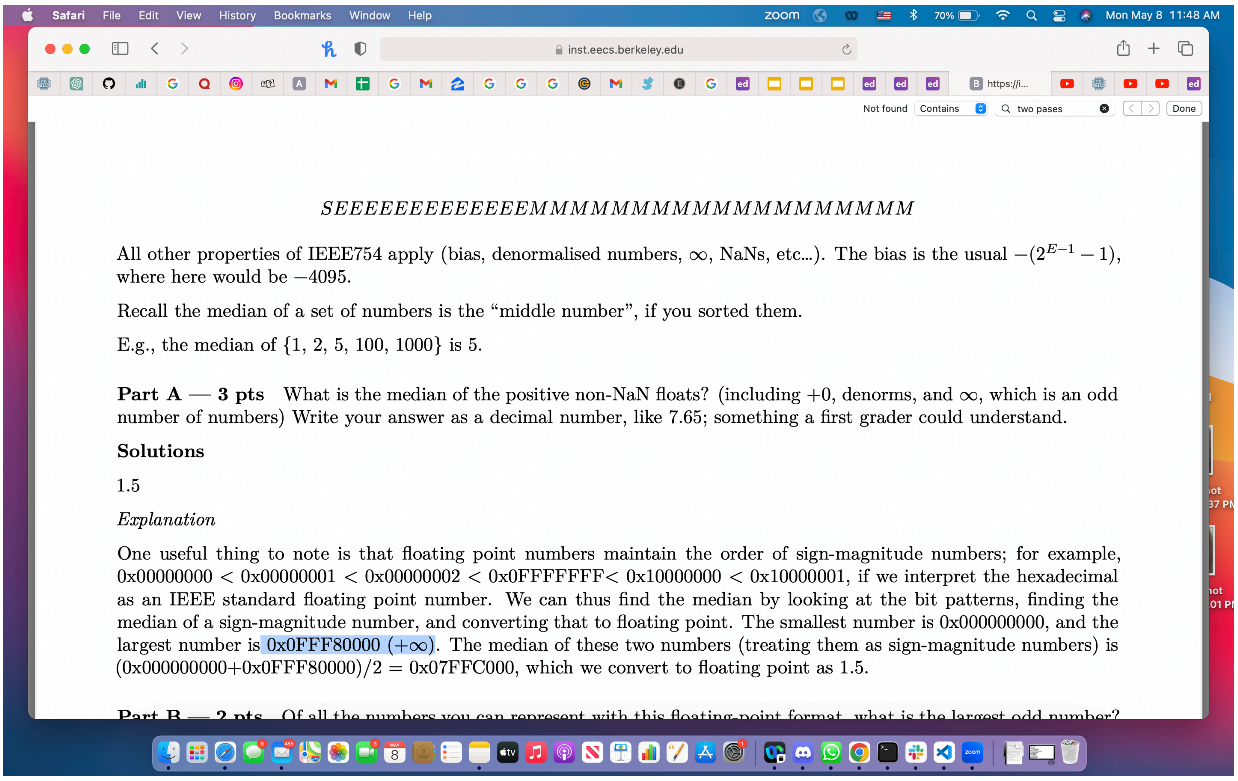
If the cache stores 4 ints per block, should access to A[2] and A[3] be hits because A[0] and A[1] stored the values?

Rosalie Fang ADMIN 2y #1700abf

#1700df

Anonymous Beaver 2y #1700aab ✓ Resolved

why is the largest number 0x0FFF80000, this does not make the exponent bits all 1s?



SEEEEEEEEEEEEEMMMMMMMMMMMMMMMMMM

All other properties of IEEE754 apply (bias, denormalised numbers, ∞, NaNs, etc...). The bias is the usual $-(2^{E-1}-1)$, where here would be -4095.

Recall the median of a set of numbers is the "middle number", if you sorted them.

E.g., the median of $\{1, 2, 5, 100, 1000\}$ is 5.

Part A — 3 pts What is the median of the positive non-NaN floats? (including $+0$, denorms, and ∞, which is an odd number of numbers) Write your answer as a decimal number, like 7.65; something a first grader could understand.

Solutions

1.5

Explanation

One useful thing to note is that floating point numbers maintain the order of sign-magnitude numbers; for example, 0x00000000 < 0x00000001 < 0x00000002 < 0x0FFFFFFF< 0x10000000 < 0x10000001, if we interpret the hexadecimal as an IEEE standard floating point number. We can thus find the median by looking at the bit patterns, finding the median of a sign-magnitude number, and converting that to floating point. The smallest number is 0x000000000, and the largest number is 0x0FFF80000 $(+\infty)$. The median of these two numbers (treating them as sign-magnitude numbers) is (0x000000000+0x0FFF80000)/2 = 0x07FFC000, which we convert to floating point as 1.5.

Part B — 2 pts Of all the numbers you can represent with this floating-point format, what is the largest odd number?

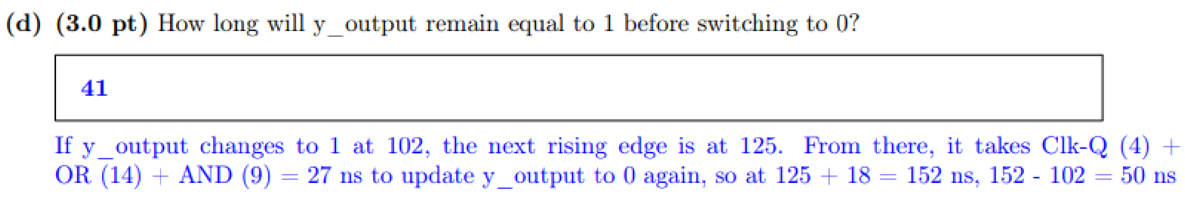
Rosalie Fang ADMIN 2y #1700abd

Yes. It said "non-NaN floats" which included infinity. The exponent bits are all 1s.

Anonymous Mink 2y #1700aaa ✓ Resolved

SU20-final-Q4D

Is the answer supposed to be 50 ns here? Not sure where 41 is coming from considering the explanation seems to say it's 50 ns.



(d) (3.0 pt) How long will y_output remain equal to 1 before switching to 0?

$\boxed{41}$

If y_output changes to 1 at 102, the next rising edge is at 125. From there, it takes Clk-Q (4) + OR (14) + AND (9) = 27 ns to update y_output to 0 again, so at 125 + 18 = 152 ns, 152 - 102 = 50 ns

1

Anonymous Emu 2y #1700aad

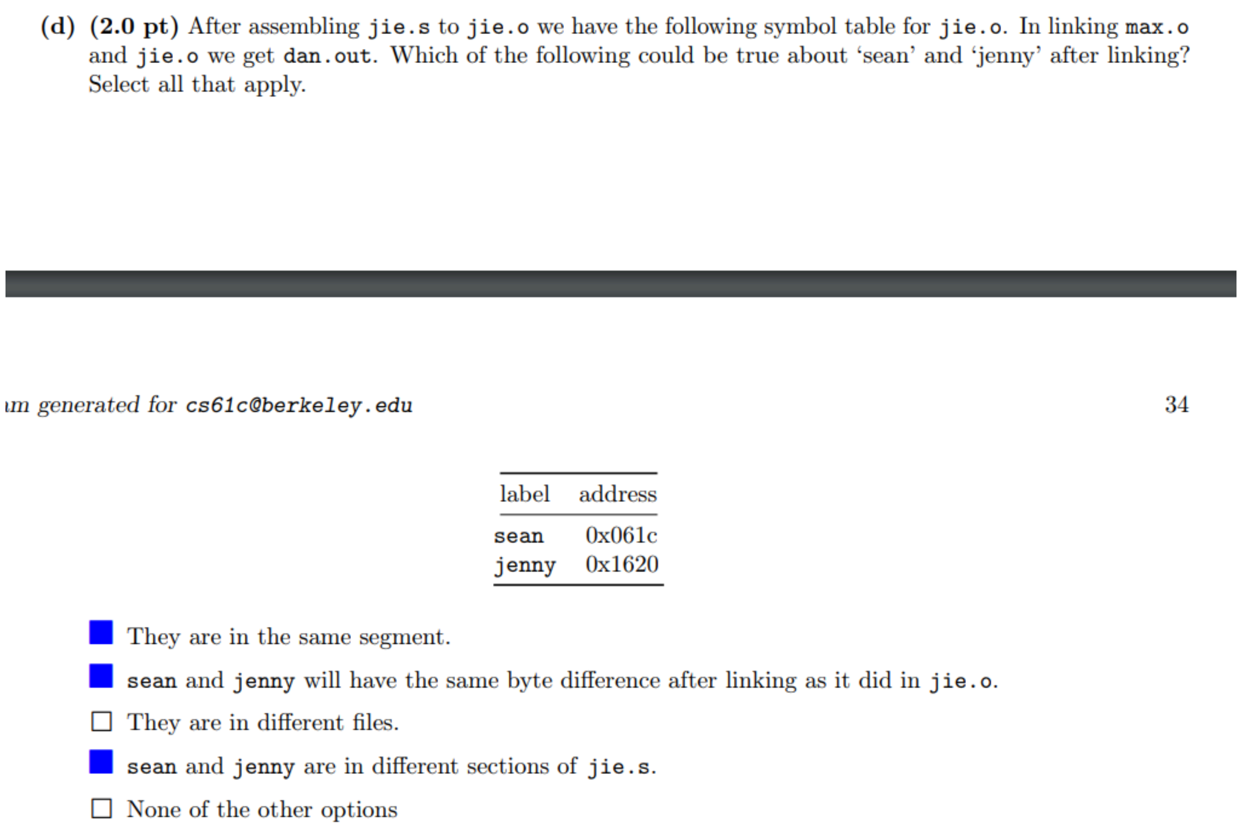
^^^

Rosalie Fang ADMIN 2y #1700abc

yeah probably typo.

Anonymous Bee 2y #1700fa Unresolved

Could someone explain why this is the answer? I'm don't really understand what the table tells us after linking



(d) **(2.0 pt)** After assembling `jie.s` to `jie.o` we have the following symbol table for `jie.o`. In linking `max.o` and `jie.o` we get `dan.out`. Which of the following could be true about 'sean' and 'jenny' after linking? Select all that apply.

um generated for ***cs61c@berkeley.edu*** 34

label	address
`sean`	0x061c
`jenny`	0x1620

■ They are in the same segment.

■ `sean` and `jenny` will have the same byte difference after linking as it did in `jie.o`.

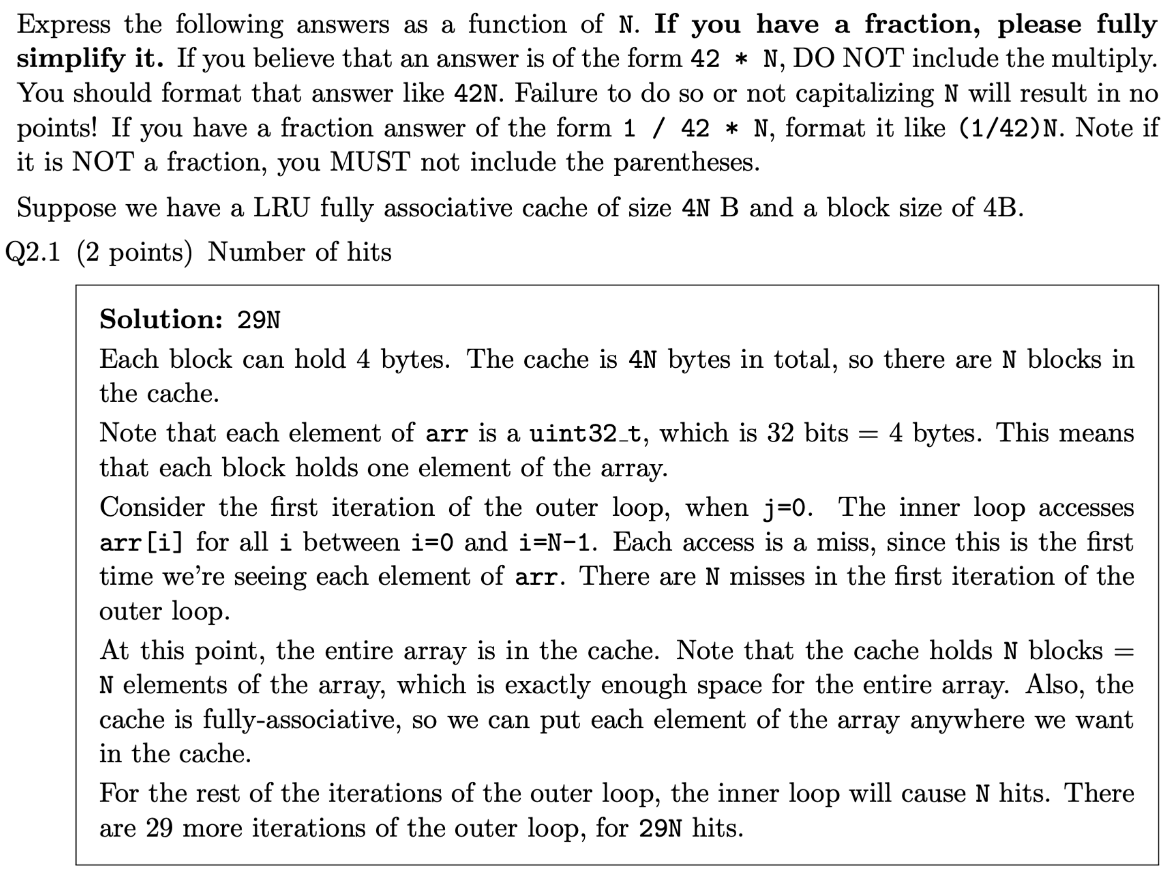
☐ They are in different files.

■ `sean` and `jenny` are in different sections of `jie.s`.

☐ None of the other options

♡ 1 ···

Anonymous Newt 2y #1700ef ✓ Resolved



Express the following answers as a function of `N`. **If you have a fraction, please fully simplify it.** If you believe that an answer is of the form `42 * N`, DO NOT include the multiply. You should format that answer like `42N`. Failure to do so or not capitalizing `N` will result in no points! If you have a fraction answer of the form `1 / 42 * N`, format it like `(1/42)N`. Note if it is NOT a fraction, you MUST not include the parentheses.

Suppose we have a LRU fully associative cache of size `4N` B and a block size of 4B.

Q2.1 (2 points) Number of hits

Solution: `29N`

Each block can hold 4 bytes. The cache is `4N` bytes in total, so there are `N` blocks in the cache.

Note that each element of `arr` is a `uint32_t`, which is 32 bits = 4 bytes. This means that each block holds one element of the array.

Consider the first iteration of the outer loop, when `j=0`. The inner loop accesses `arr[i]` for all `i` between `i=0` and `i=N-1`. Each access is a miss, since this is the first time we're seeing each element of `arr`. There are `N` misses in the first iteration of the outer loop.

At this point, the entire array is in the cache. Note that the cache holds `N` blocks = `N` elements of the array, which is exactly enough space for the entire array. Also, the cache is fully-associative, so we can put each element of the array anywhere we want in the cache.

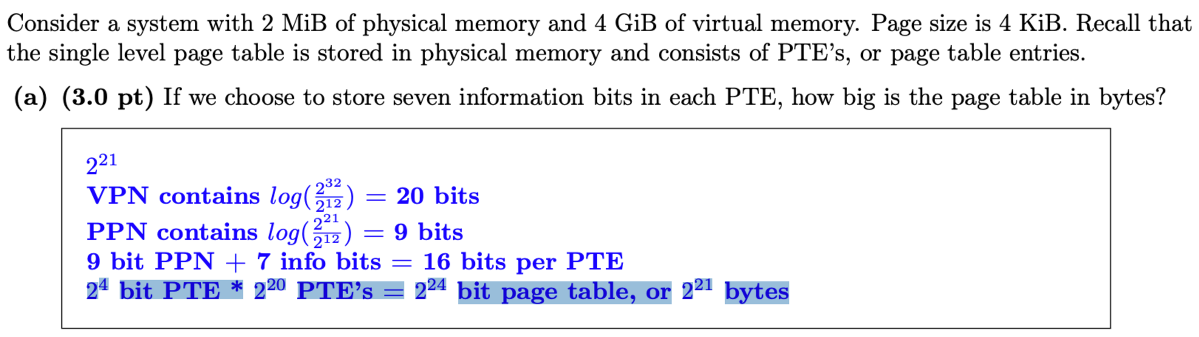
For the rest of the iterations of the outer loop, the inner loop will cause `N` hits. There are 29 more iterations of the outer loop, for `29N` hits.

sp20 q2.1: how is it loading the entire array in the cache when the cache has a total size of 4N B? Supposed B was 1, wouldn't it require to be a miss at evvery access? Could someone explain the reasoning behind this soltuion

R Rosalie Fang ADMIN 2y #1700ff

What do you mean B was 1? The cache is 4N bytes with N being the length of the array. there are 4 bytes per integer so 4N is exactly the number of bytes in the array.

Anonymous Snail 2y #1700ea ✓ Resolved


Consider a system with 2 MiB of physical memory and 4 GiB of virtual memory. Page size is 4 KiB. Recall that the single level page table is stored in physical memory and consists of PTE's, or page table entries.

(a) **(3.0 pt)** If we choose to store seven information bits in each PTE, how big is the page table in bytes?

2^{21}

VPN contains $log(\frac{2^{32}}{2^{12}})$ **= 20 bits**

PPN contains $log(\frac{2^{21}}{2^{12}})$ **= 9 bits**

9 bit PPN + 7 info bits = 16 bits per PTE

2^4 **bit PTE *** 2^{20} **PTE's =** 2^{24} **bit page table, or** 2^{21} **bytes**

sp20-final-q4.1

what is the explanation for the last part?

R Rosalie Fang ADMIN 2y #1700fe

There are 7 information bits in each PTE, and 9 PPN bits, hence 16 bits total of PTE (2^4). Then there are 2^20 PTEs because that's the total number of virtual pages, (virtual pages index PTEs).

Ayati Sharma 2y #1700df ✓ Resolved

Su20 Q6. Is it possible to get a full explanation for this question? How is A[2] read a miss for processor 0 if we loaded in A[0]-A[3] when reading A[0]?

1

Anonymous Koala 2y #1700fb

having the same question!

R Rosalie Fang ADMIN 2y #1700fd

The accesses are interleaved, so A[0], A[1], A[2], A[3], and process 0 handles only even indices. Hence, A[0] draws in the block A[0]-A[3] for processor 0, however, right after A[1] draws in the block A[0]-A[3] for processor 1 and effectively disabling access for processor 0 due to MOESI (since processor 1 writes to A[1]). Hence when A[2] tries to read again it misses.

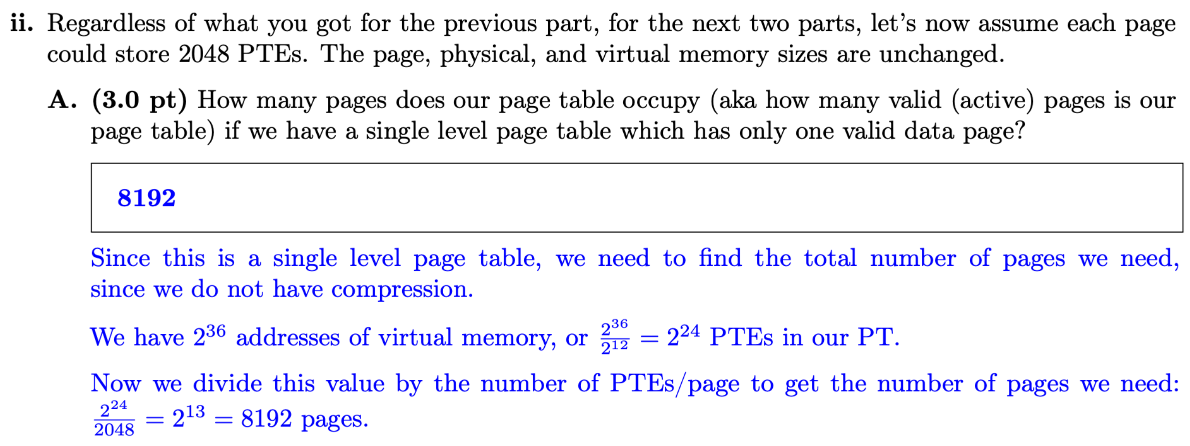
1

Anonymous Emu 2y #1700aae

so the cache is only flushed after writing?

Stella Kaval 2y #1700de ✓ Resolved

Summer 2020 1. It's All an Illusion: I am confused how they got to this solution



ii. Regardless of what you got for the previous part, for the next two parts, let's now assume each page could store 2048 PTEs. The page, physical, and virtual memory sizes are unchanged.

A. **(3.0 pt)** How many pages does our page table occupy (aka how many valid (active) pages is our page table) if we have a single level page table which has only one valid data page?

8192

Since this is a single level page table, we need to find the total number of pages we need, since we do not have compression.

We have 2^{36} addresses of virtual memory, or $\frac{2^{36}}{2^{12}} = 2^{24}$ PTEs in our PT.

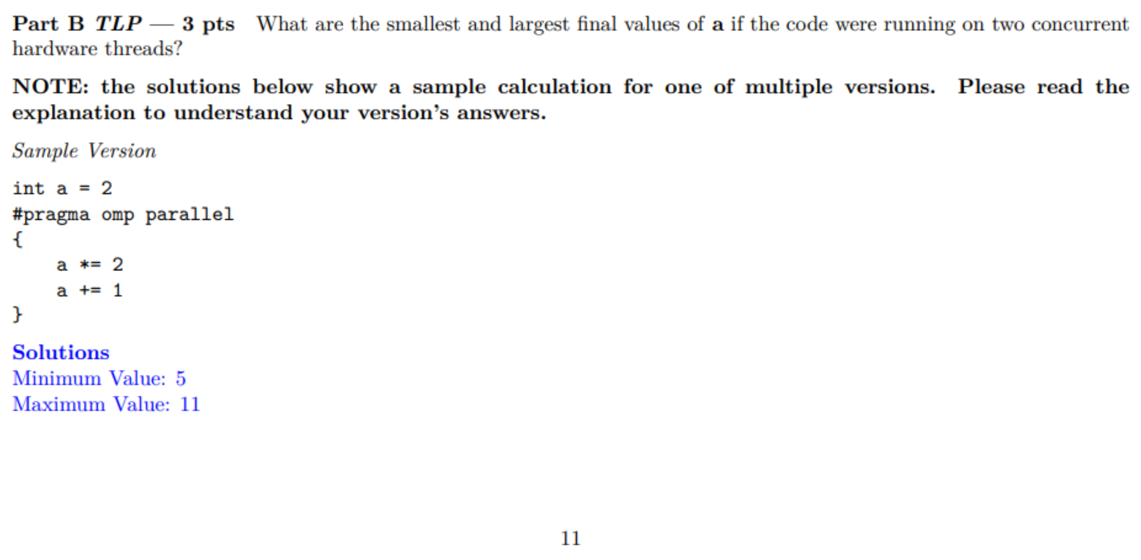
Now we divide this value by the number of PTEs/page to get the number of pages we need: $\frac{2^{24}}{2048} = 2^{13} = 8192$ pages.

Rosalie Fang ADMIN 2y #1700ed

There are 36 bits of virtual address and 12 bits of offset. That means that there are 24 bits of VPN. VPN is used to index into the page table, so since there are 2^24 unique VPN values there are 2^24 PTEs in our page table. Now the number of pages in the page table is just dependent on how many entries we can fit into a page. hence the 2^24/2048.

Anonymous Aardvark 2y #1700dd ✓ Resolved

Fa20-Final-Sec3ParallelismB:



Part B ***TLP*** **— 3 pts** What are the smallest and largest final values of **a** if the code were running on two concurrent hardware threads?

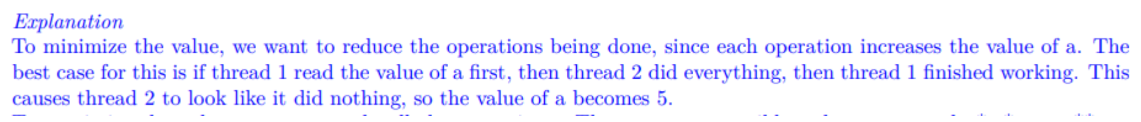
NOTE: the solutions below show a sample calculation for one of multiple versions. Please read the explanation to understand your version's answers.

Sample Version

```
int a = 2
#pragma omp parallel
{
    a *= 2
    a += 1
}
```

Solutions
Minimum Value: 5
Maximum Value: 11

11



Explanation
To minimize the value, we want to reduce the operations being done, since each operation increases the value of a. The best case for this is if thread 1 read the value of a first, then thread 2 did everything, then thread 1 finished working. This causes thread 2 to look like it did nothing, so the value of a becomes 5.

1. Can someone clarify the min-val case? The way TLP was conceptualized in the lecture had us write out a complete set of instructions for both. So in TLP, we don't have to execute all the instructions for each thread, is that correct? One thread can jal Exit before another thread finishes?

2. If that's the case, why did the solution take the extra work for Thread 2 to do anything? Wouldn't another possible answer be: thread 1 completes all of the instructions and exits; Thread 2 doesn't do anything? As I see it, this single-thread answer is a subset of TLP, right?

3. Also, I know the lecture recommends writing out the code in assembly, so TLP operates at assembly-level granularity, is that correct? Or can you potentially just do it in C for problems like this (I think it works out the same here)?

R Rosalie Fang ADMIN 2y #1700ec

1. We do have a execute all instructions for each thread. The problem here is the very common data-race problem of read-modify-write. The sequence of code in risc-v will be (read a, a*2, write a, read a, a+1, write a). a starts as 2. Let's say thread 1 executes until almost finish (read a = 2, a*2, write a = 4, read a = 4, a + 1) and then switched to thread 2 and thread 2 executes to finish (read a = 4, a*2, write a = 8, read a = 8, a+1, write a = 9). and then we switch back to thread 1 to finish up so it writes a=5 because that's what it knows since it was in the middle of the line `a+=1` so it only know the old value of a.

2. We do have a execute all instructions for each thread.

3. Yes, you always operate at assembly-level granularity. You can write out pseudocode by remember whenever you have something like `+=`, or `a = a*2` , there's always the read, then the compute, and then the write back.

1

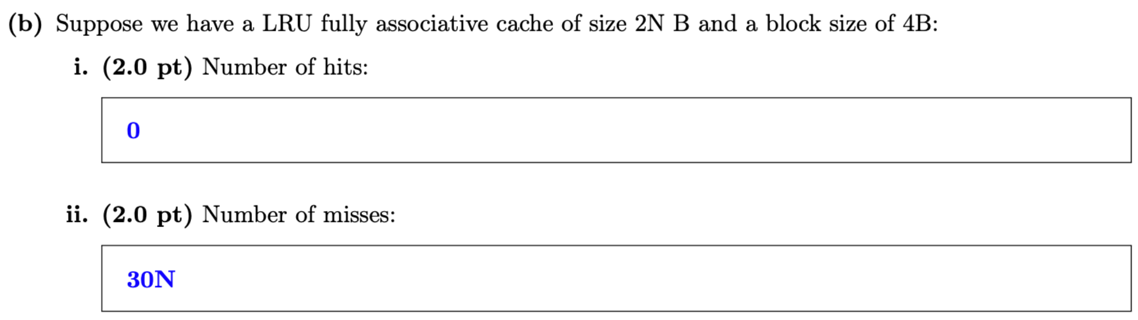
Anonymous Aardvark 2y #1700ee

Thanks, Rosalie!

Anonymous Bee 2y #1700cd ✓ Resolved

sp20 final

how come the answers changed from the previous part?



(b) Suppose we have a LRU fully associative cache of size 2N B and a block size of 4B:

i. (2.0 pt) Number of hits:

0

ii. (2.0 pt) Number of misses:

30N

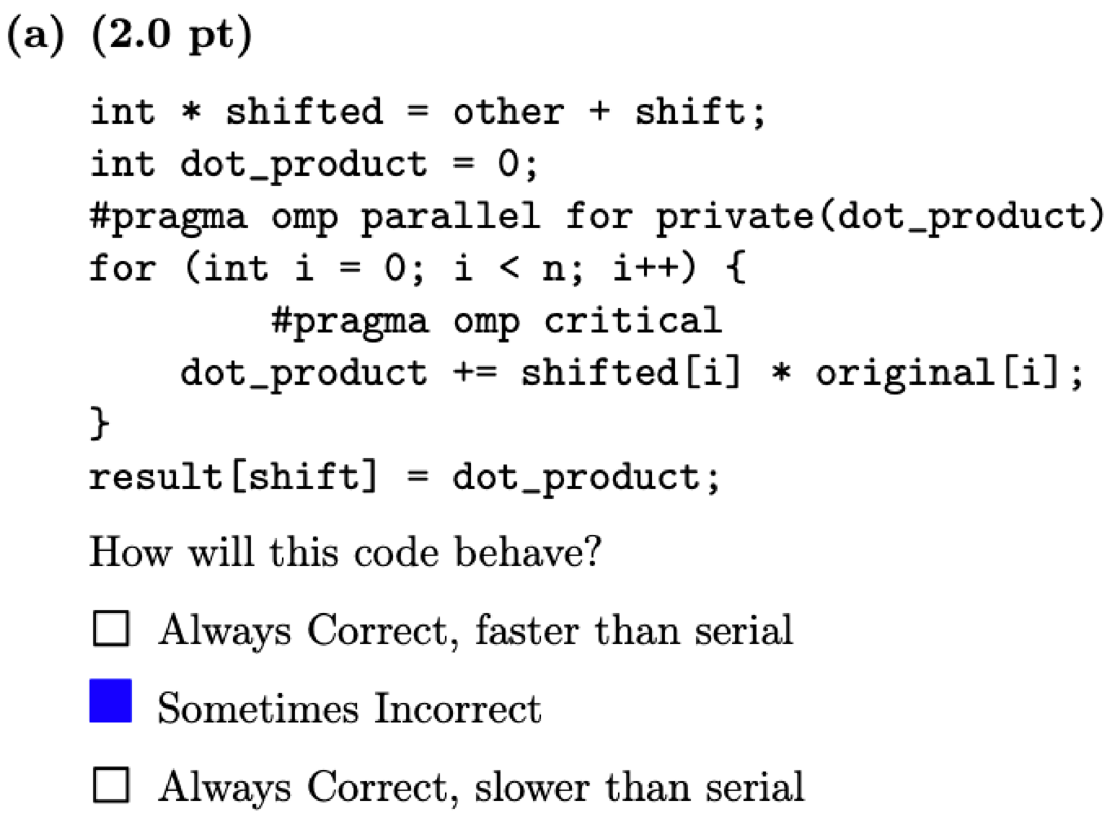
Rosalie Fang ADMIN 2y #1700eb

The difference is that the cache size shrunk in half. So in the inner for loop, arr[0] - arr[N/2] get loaded and then replaced by arr[N/2 + 1] - arr[N - 1], and that process repeats.

Anonymous Hippopotamus 2y #1700cc ✓ Resolved

SU20-final-Q2

Why is (a) sometimes incorrect and (b) always correct but slower than serial?



(a) (2.0 pt)

```
int * shifted = other + shift;
int dot_product = 0;
#pragma omp parallel for private(dot_product)
for (int i = 0; i < n; i++) {
        #pragma omp critical
    dot_product += shifted[i] * original[i];
}
result[shift] = dot_product;
```

How will this code behave?

☐ Always Correct, faster than serial

[✓ Sometimes Incorrect]

☐ Always Correct, slower than serial

(b) (2.0 pt)

```
int * shifted = other + shift;
int dot_product = 0;
#pragma omp parallel for reduction(+:dot_product)
for (int i = 0; i < n; i++) {
        #pragma omp critical
    dot_product += shifted[i] * original[i];
}
result[shift] = dot_product;
```

How will this code behave?

[✓ Always Correct, slower than serial]

☐ Always Correct, faster than serial

☐ Sometimes Incorrect

♡ 1 ···

R Rosalie Fang ADMIN 2y #1700fc

(a) sometimes incorrect because of the private(dot_product) declared dot_product to be private to each thread but then has, after the thread finishes, result[shift] = dot_product, so it'd only be correct if the actual dot product happened to be the same value of the dot_product computed by one of the threads.

(b) is correct but slower than serial because the only thing that's happening in the loop is forced to be serialized due to #pragma omp critical, so none of the threads are taking advantage of the multi-threading aspect since all of them have to go in order anyways. Therefore the overhead created by the threads actually bring overall disadvantage hence slower than serial

♡ ···

Anonymous Bee 2y #1700cb ✓ Resolved

su20-final

for q4 part c, how did they get the final answer? I'm not sure why the numbers in the answer key were added

♡ ···

R Rosalie Fang ADMIN 2y #1700da

The first cycle starts at 25ns because that's the first rising edge of the clock, but since x = 0, y_output is 0. Then the clock rises, x changes 30s after the rising edge of the clock, so for the first full cycle nothing changes. y_output is still 0. The next rising edge of the clock is an entire clock cycle away, so 25 + 50ns, now we know x = 1. Reg 0 should switch to be 1 after clock-2-q delay, and Reg 1 should remain 0 since it's the result of the previous y_output AND x which is 0. So starting at Reg0 and Reg1, we need to account for the clk-2-q delay which is 4ns, and then the longest combinational logic delay is max(NOT, OR) + AND which is OR + AND = 14 + 9 = 23 ns. Therefore it's 75 + 4 + 23 = 102.

1 ...

Anonymous Bee 2y #1700db

thank you so much!

...

Anonymous Rhinoceros 2y #1700bf ✓ Resolved

su20-final-q1biA): how is the number of PTEs = 2^6?

B): how do u calculate the #L2 PTs and # PTEs in L2?

...

R Rosalie Fang ADMIN 2y #1700cf

Number of PTE = 2^(VPN bits). This question specified a 2-level page table which is out of scope.

1 ...

Anonymous Dunlin 2y #1700be Unresolved

SU20_Final_Q7, I am confused on part d. What is the difference between segment and section in answer choices 1 and 4? And why wouldn't it be possible that one of sean and jenny is in the static segment and the other is in the text segement? Thanks!

...

Anonymous Dunlin 2y #1700bd ✓ Resolved

SU20-Final-Q6, part f, do we write back the data that we modified to memory if we evict blocks due to coherence misses? The wouldn't we be writing back 32 times, why is the answer 0? Thanks!

...

R Rosalie Fang ADMIN 2y #1700ce

We "snoop" another cache for data, that doesn't mean we have to write back to main memory. Write-back is only needed when we evict a block to memory. For coherency misses, we don't need to evict blocks since it's not getting replaced, we only change the status bits.

1 ...

Anonymous Raccoon 2y #1700abb

So when proc 1 loads in the A[1] block, does it immediately get the changes proc 0 made to it?

...

R Rosalie Fang ADMIN 2y #1700abe

Replying to Anonymous Raccoon

Yes.

...

Anonymous Elephant 2y #1700af ✓ Resolved

Su20-Final-Q9cii

If we have 7 parity bits, that means the greatest parity bit is p64. Doesn't this mean the bits of data we can cover is equal to 2^6 - 1 + 7 = 70?

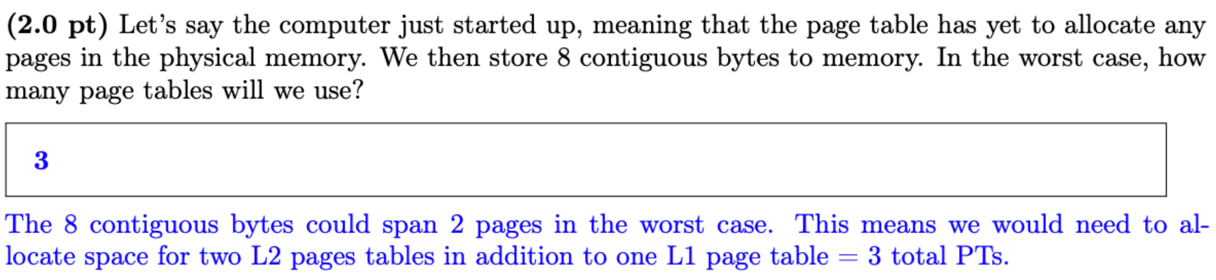
...

J Jero Wang ADMIN 2y #1700bb

7 parity bits means that we have 2^7-1 total bits, and we must subtract the 7 parity bits, so we have 120 total.

Anonymous Eagle 2y #1700aa ✓ Resolved

Su20-Final-Q1bii


(2.0 pt) Let's say the computer just started up, meaning that the page table has yet to allocate any pages in the physical memory. We then store 8 contiguous bytes to memory. In the worst case, how many page tables will we use?

3

The 8 contiguous bytes could span 2 pages in the worst case. This means we would need to allocate space for two L2 pages tables in addition to one L1 page table = 3 total PTs.

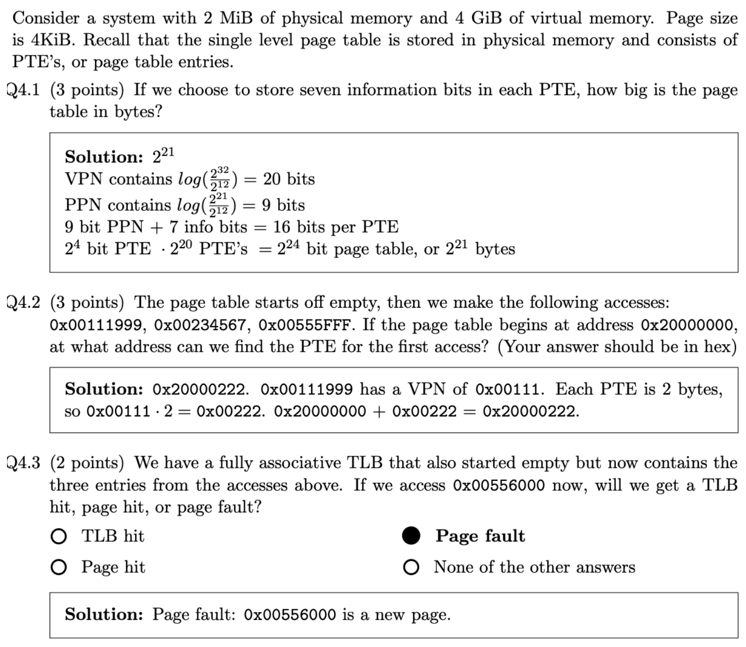
I'm not sure why 8 contiguous bytes span 2 pages given that page size is 4KiB.

E Erik Yang TA 2y #1700ae

If some of the 8 bytes stored at the end of 1 page, it will spill into another. And those 2 data pages require 1 L2 page each. So 2 L2 pages for 1 L1 page table means 3 PT total

Anonymous Eagle 2y #1700f ✓ Resolved

Sp20-Final-Q4.3


Consider a system with 2 MiB of physical memory and 4 GiB of virtual memory. Page size is 4KiB. Recall that the single level page table is stored in physical memory and consists of PTE's, or page table entries.

Q4.1 (3 points) If we choose to store seven information bits in each PTE, how big is the page table in bytes?

Solution: 2^{21}
VPN contains $log(\frac{2^{32}}{2^{12}}) = 20$ bits
PPN contains $log(\frac{2^{21}}{2^{12}}) = 9$ bits
9 bit PPN + 7 info bits = 16 bits per PTE
2^4 bit PTE $\cdot 2^{20}$ PTE's $= 2^{24}$ bit page table, or 2^{21} bytes

Q4.2 (3 points) The page table starts off empty, then we make the following accesses: 0x00111999, 0x00234567, 0x00555FFF. If the page table begins at address 0x20000000, at what address can we find the PTE for the first access? (Your answer should be in hex)

Solution: 0x20000222. 0x00111999 has a VPN of 0x00111. Each PTE is 2 bytes, so 0x00111 · 2 = 0x00222. 0x20000000 + 0x00222 = 0x20000222.

Q4.3 (2 points) We have a fully associative TLB that also started empty but now contains the three entries from the accesses above. If we access 0x00556000 now, will we get a TLB hit, page hit, or page fault?

○ TLB hit [✓ **Page fault**]
○ Page hit ○ None of the other answers

Solution: Page fault: 0x00556000 is a new page.

I understand 0x00556000 is not in TLB due to index difference, but could you explain why it's a new page?

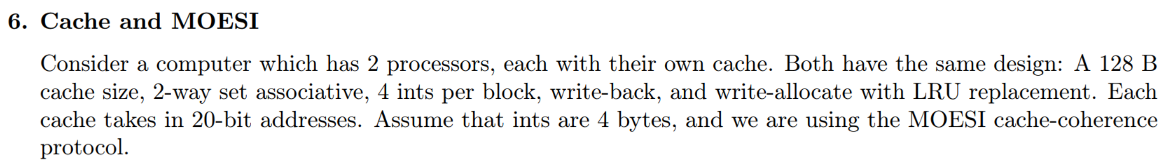
J Jero Wang ADMIN 2y #1700ba

New page here is referring to the fact that none of the entries in the TLB has a matching index, and it is a page new to the TLB.

Anonymous Pelican 2y #1700e ✓ Resolved

Su20 final 6

Is cache coherence in scope?



6. Cache and MOESI

Consider a computer which has 2 processors, each with their own cache. Both have the same design: A 128 B cache size, 2-way set associative, 4 ints per block, write-back, and write-allocate with LRU replacement. Each cache takes in 20-bit addresses. Assume that ints are 4 bytes, and we are using the MOESI cache-coherence protocol.

Erik Yang TA 2y #1700ac

moesi is in scope

Anonymous Fox 2y #1700d ✓ Resolved

SU20-Final-Q1.b

Are hierarchial page tables in scope?

Erik Yang TA 2y #1700ab

nope 2 level page tables are not in scope

Anonymous Aardvark 2y #1700c ✓ Resolved

SU20-MT2-Q1.iiB

For this question, my thought process was first figuring out how many entries in each page table.

For L1: it we would have 2^12 entries as we have 12 VPN1 bits. We know that each page stores 2^11 PTE's (which is given in the problem statement) so for this we would need 2 pages.

For l2: we know that the number of page tables is going to be the number of page table entries for L1 page table which is 2^12 ==> so we have 2^12 page tables for l2? and we know that for each page table we would need two pages

so wouldn't the answer be 2 (L1)+ 2^12*2 (l2)= 2 + 2^13?

I think my reasoning is going to wrong when trying to figure out how many pages L2 needs. I'm a bit confused with the written solutions so if you could clarify that would be great!

Jero Wang ADMIN 2y #1700bc

Since there's only one data page, you would only need one L2 page table. This is one of the benefits of having a multi-level page table, since you only allocate parts of it as needed.

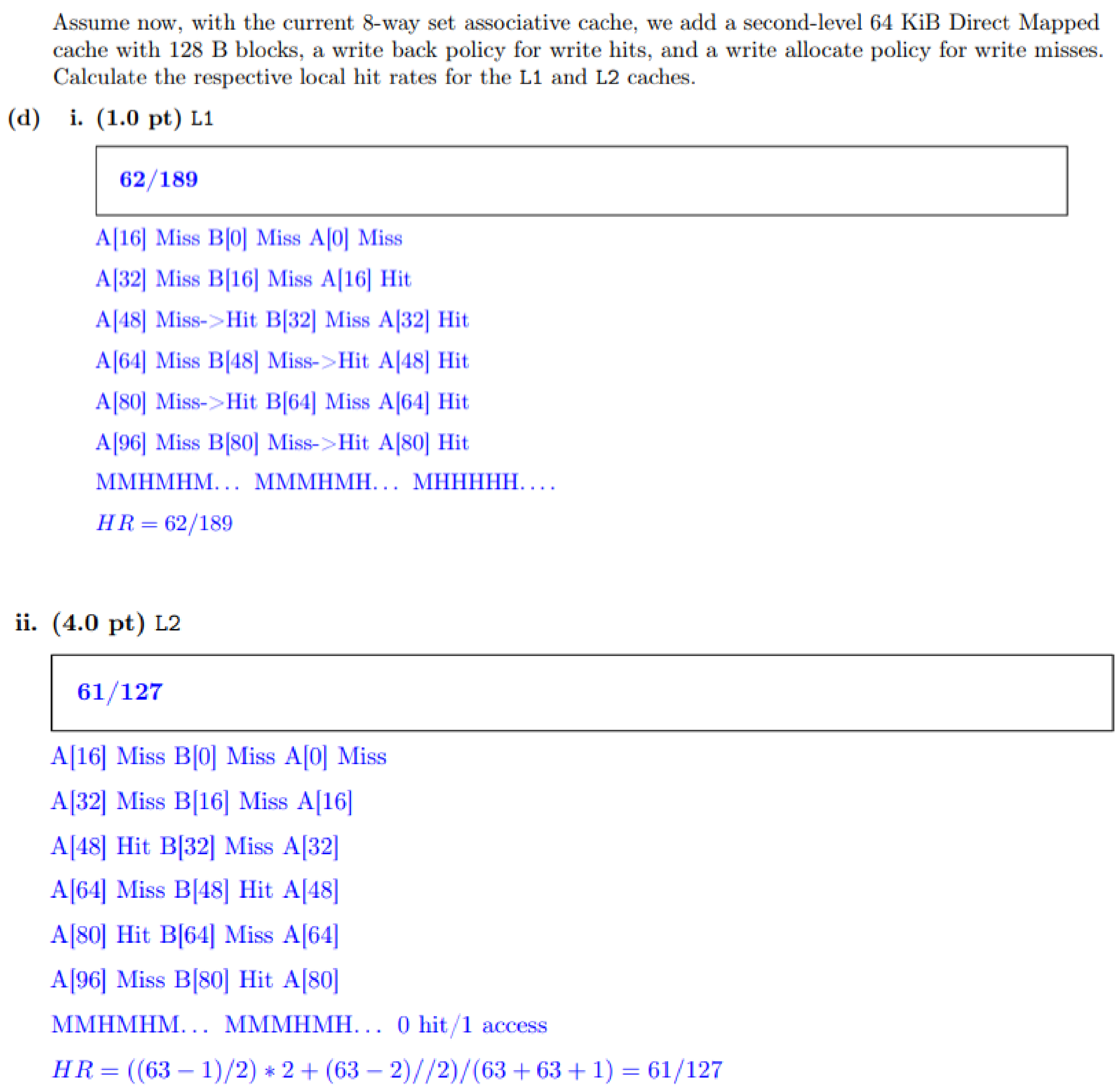
However, this is not in scope this semester.

1

Anonymous Mongoose 2y #1700b Unresolved

Su20 MT2 4d

Could someone provide a more detailed explanation of how we got these answers? Thank you



Assume now, with the current 8-way set associative cache, we add a second-level 64 KiB Direct Mapped cache with 128 B blocks, a write back policy for write hits, and a write allocate policy for write misses. Calculate the respective local hit rates for the L1 and L2 caches.

(d) **i. (1.0 pt)** L1

$\boxed{62/189}$

A[16] Miss B[0] Miss A[0] Miss

A[32] Miss B[16] Miss A[16] Hit

A[48] Miss->Hit B[32] Miss A[32] Hit

A[64] Miss B[48] Miss->Hit A[48] Hit

A[80] Miss->Hit B[64] Miss A[64] Hit

A[96] Miss B[80] Miss->Hit A[80] Hit

MMHMHM... MMMHMH... MHHHHH....

$HR = 62/189$

ii. (4.0 pt) L2

$\boxed{61/127}$

A[16] Miss B[0] Miss A[0] Miss

A[32] Miss B[16] Miss A[16]

A[48] Hit B[32] Miss A[32]

A[64] Miss B[48] Hit A[48]

A[80] Hit B[64] Miss A[64]

A[96] Miss B[80] Hit A[80]

MMHMHM... MMMHMH... 0 hit/1 access

$HR = ((63 - 1)/2) * 2 + (63 - 2)//2)/(63 + 63 + 1) = 61/127$

Anonymous Mongoose 2y #1700a ✓ Resolved

Is there by any chance a walkthrough to SU20 MT2? Thank you

E Erik Yang TA 2y #1700ad

unfortunately, i couldn't find one either

1