B461 A8

by Matthew Ploetz

1

A)

i. $4096 = M \cdot 12 + (M \cdot 1) \cdot 9$

$M = 195$

$\text{Log}_{195} 100000000 = 3.49 \sim 3.5 = \#\text{Nodes to be examined}$

$10\text{ms} \cdot \text{ceiling}(3.5) = \mathbf{40ms}$

ii. $M = 195$

$\text{Max} = \log_{\frac{195+1}{2}}(\frac{100000000}{2}) + 1$

$= 4.89 \sim 4.9$

$\text{ceiling}(4.9) = 5 \cdot 10\text{ms} = \mathbf{50ms}$

iii. $7 = \log_{\frac{195+1}{2}}\left(\frac{N}{2}\right) + 1$

$\boldsymbol{N = 1.74 \cdot 10^{14}}$

iv. It would change our branch factor so we would have:

$8192 = M \cdot 12 + (M \cdot 1) \cdot 9$

$M = 389.7$

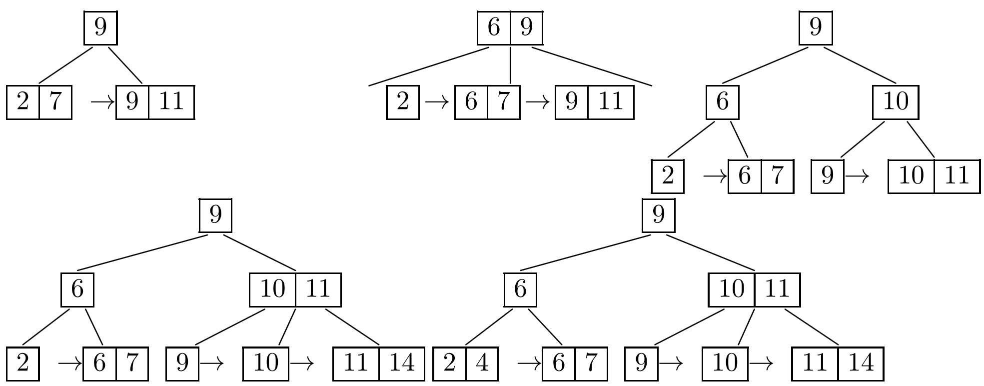
$\text{Max} = \log_{\frac{389.7}{2}}(\frac{100000000}{2}) + 1$

$= 4.36 \sim 4.4 = \text{ceiling}(4.4) = 5 \cdot 10\text{ms} = \mathbf{50ms}$

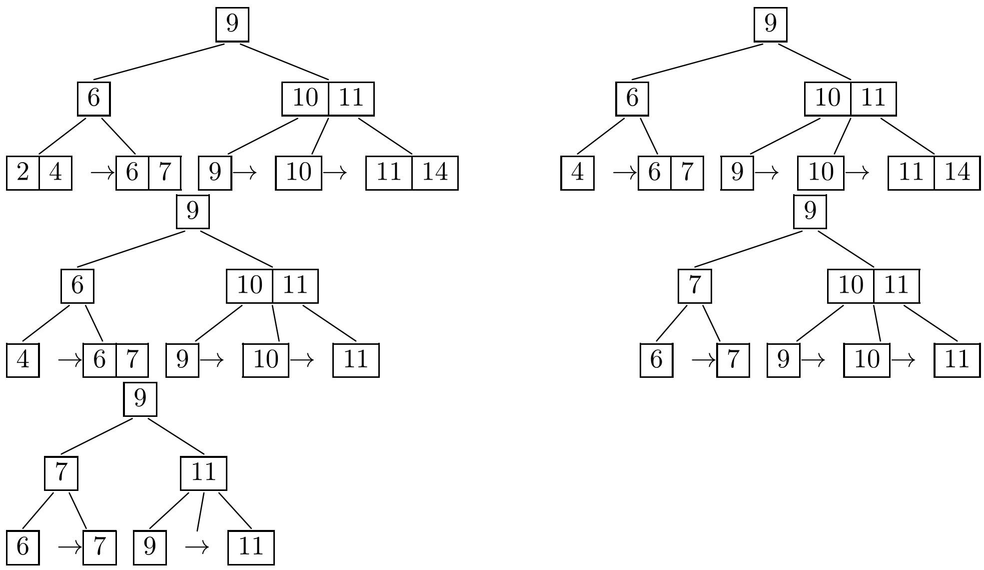
Therefore, this block size would not change our max access time.

B)

i.



ii.



2

A)

i.

a)

b)

c)

d)

ii.

a)

b)

c)

3

A) $B(R) = 1500000/30$, $B(S) = 5000/10 = M = 101$ So using the nested loop alg we get $B(S) + (B(R)B(S)/M - 1).500 + (50000 \cdot 500/101 - 1) =$ **250500 block IO's**

B) $5 \cdot (B(R) + B(S)) = 5 \cdot (50000 + 500) = \mathbf{252500\ block\ IO's}$. M1>M2 101 > sqrt(min ($B(R)$, $B(S)$)) $\sim 101 \ngtr 500$. But we ignore this as instructed.

C) $3 \cdot (50000 + 500) = \mathbf{151500\ block\ IO's}$

4

A) Yes, R1(z)R2(x)R1(x)R2(y)R1(y) eqv.

B) Not serializable. Because we write at W2(y) and later come back to read at R1(y) so it is not in chronological order and is a cycle which we cannot have for a schedule.

C) No, because we wrote to W2(x) but did not read to it after it before writing again with W1(x). So its overwritten and value was not locked.

5

Let T1=R1(A) W1(A) R1(B) W1(B)

T2=R2(A) W2(A) R2(B) W2(B)

T3=R3(B) W3(B) R3(A) W3(A)

S=R1(A) W1(A) R2(A) W2(A) R1(B) W1(B) R2(B) W2(B) R3(B) W3(B) R3(A) W3(A)

S1=R1(B) W1(B) R1(A) W1(A) R2(A) W2(A) R2(B) W2(B) R3(B) W3(B) R3(A) W3(A)

S2= R2(A) W2(A) R1(A) W1(A) R1(B) W1(B) R2(B) W2(B) R3(B) W3(B) R3(A) W3(A)

6

Let T1=R1(A) W1(A) R1(B) W1(B)

T2=R2(A) W2(A) R2(B) W2(B)

T3=R3(B) W3(B) R3(A) W3(A)

S=R1(A) W1(A) R2(A) W2(A) R1(B) W1(B) R2(B) W2(B) R3(B) W3(B) R3(A) W3(A)

Yes, because since there are no conflictinig pairs then any number of non conflicting swaps and adjacent swaps can transform them to be serial schedules.

7

S=R1(A) W1(A) R1(B) W1(B) R1(C) W1(C) R2(A) R2(B) W2(A) W2(B) R2(C) R3(C) W3(A) W3(B) W3(C)

T1=R1(A)W1(A)R1(B)W1(B)R1(C)W1(C)

T2=R2(A)R2(B)W2(A)W2(B)R2(C)W2(C)

T3=R3(A)R3(B)W3(A)W3(B)R3(C)W3(C)

8

A) For schedule (T_1, T_2), $A = 0$ and $B = 0$. After first transaction $B = 1$, since $A = 0$ and we write to B. So we have $A=0$ and $B=1$. Then we do transaction2 and nothing happens because B is not 0. So we end up with $A=0$ and $B=1$, thus since at least one of the values are 0 the the consistency requirement is met.

For (T_2, T_1) $A=0$ and $B=0$, then after T_2 we get $A=1$ because $B=0$ and we increment and write to A. Then we go to T_1 and since $A \neq 0$ nothing changes. So we end up with $A = 1$ and $B = 1$, thus since we have at least on e of the values are 0 then the consistency requirement is met.

B) $S = R1(A)\, R2(B)\, R1(B)\, R2(A)\, W1(B)\, W2(A)$

C) No, because each serial schedule results with T_1 action and T_2 action conflict with each other, therefore there cannot be a schedule made $\text{non} - \text{serial}$ that will be a serializable schedule.