CPSC 221
Theory Assignment 2

Date Mitchell Chan, c7y9a
Sandra Yoo a7x9a

1) Let's say that x^5 is $O(x^2)$. If this is true, then there exists positive constants c & n_0 such that for all $n \geq n_0$, $n^5 \leq cn^2$

$$n^5 \leq cn^2$$
$$n^3 \leq c$$

This fails for n greater than c. Therefore we have a contradiction and x^5 is not $O(x^2)$.

2) $\sum_{i=1}^{n} i^3 = \frac{n^2(n+1)^2}{4} = \frac{n^2(n^2+2n+1)}{4} = \frac{n^4+2n^3+n^2}{4}$

By the definition of Big-Θ, the runtime is $\Theta(n^4)$ if and only if its $O(n^4)$ and $\Omega(n^4)$.

Q Let $f(n) = \frac{n^4+2n^3+n^2}{4}$. $f(n) = O(g(n))$ iff $f(n) \leq cg(n)$ for all $n \geq n_0$, where c and n_0 are positive constants.

$\frac{f(n)}{g(n)} \leq c$ for all $n \geq n_0$

$$\frac{n^4+2n^3+n^2}{4} \leq cn^4$$
$$\frac{n^4}{4n^4} + \frac{2n^3}{4n^4} + \frac{n^2}{4n^4} \leq c$$
$$\frac{1}{4} + \frac{1}{2n} + \frac{1}{4n^2} \leq c$$
$$\frac{1}{4} + \frac{1}{2} + \frac{1}{4} \leq c \quad , n_0 = 1$$
$$1 \leq c \qquad \boxed{c = 1, n_0 = 1}$$

Mitchell Chan c7g6a
Sandra Yoo a9x9a

Date

2. $f(n) \geq cg(n)$ for all $n \geq n_0$ where c + n_0 are positive constants.

$\frac{f(n)}{g(n)} \geq c$ for all $n \geq n_0$

$\frac{n^4 + 2n^3 + n^2}{4} \geq cn^4$

$\frac{1}{4} + \frac{1}{2n_0} + \frac{1}{4n_0^2} \geq c$

$\frac{9}{16} \geq c$, $n_0 = 2$

$\boxed{c = \frac{1}{4}, n_0 = 2}$

Thus, since $1^3 + 2^3 + 3^3 + \ldots + n^3$ is shown to be $O(n^4)$ and $\Omega(n^4)$, we can say it is $\Theta(n^4)$. QED.

3) $E_1 = 0$

$E_k = E_{k-1} + k + 1$, $k > 1$

k		
2	$E_1 + k + 1$	0 , 3 , (3+4) , (3+4+5) + . . . +(3+4+ . . . +k+k+1)
	$= 3$	E_1 E_2 E_3 E_4 E_k
3	$E_2 + k + 1$	
	$= 7$	$\boxed{E_k = \frac{1}{2}k^2 + \frac{3}{2}k - 2}$
4	$E_3 + k + 1$	
	$= 12$	

Page

$E_1 = 0$
$E_k = E_{k-1} + k + 1$

Date

3b) Base case: $E_1 = 0 \Rightarrow \underbrace{\frac{1}{2}(1)^2 + \frac{3}{2}(1) - 2}_{\text{RHS}} = 0$

Inductive Hypothesis: $E_k = \frac{1}{2}k^2 + \frac{3}{2}k - 2$

Inductive Step:

LHS:
$E_{k+1} = E_{(k+1)+1} + (k+1) + 1$
$= E_k + k + 2$
$= \frac{1}{2}k^2 + \frac{3}{2}k - 2 + k + 2$
$= \frac{1}{2}k^2 + \frac{5}{2}k$
$= k(\frac{1}{2}k + \frac{5}{2})$

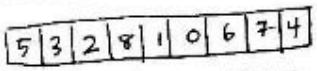
RHS:
$\frac{1}{2}(k+1)^2 + \frac{3}{2}(k+1) - 2$
$= \frac{1}{2}(k^2+2k+1) + \frac{3}{2}k + \frac{3}{2} - 2$
$= \frac{1}{2}k^2 + k + \frac{1}{2} + \frac{3}{2}k + \frac{3}{2} - 2$
$= \frac{1}{2}k^2 + \frac{5}{2}k$
$= k(\frac{1}{2}k + \frac{5}{2})$

LHS = RHS

QED.

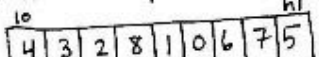
4)

MITCHELL CHAN R7Y9A
SANDRA YOO A9X9A

(#4) we start with: 

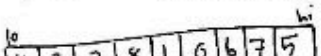
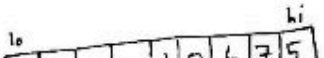
$i=0$ (5), $j=8$ (4) and $k = \frac{\lceil 0+8 \rceil}{2} = 4$ (1).

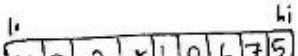
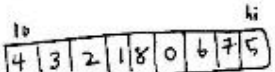
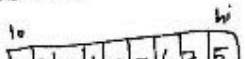
we find that 4 is the median, and so we swap this with the 1st value

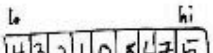
and we end up with 

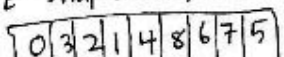
we now perform "if (x[i] < x[lo]) swap (x[++p], x[i])"

p i ↳ x[i] < x[lo], perform swap


p i ↳ x[i] < x[lo], perform swap ⇒ 
p i ↳ no swap ⇒


p i ↳ perform swap ⇒ 
p i ↳ perform swap ⇒ 
p i ↳ no swap 2x ⇒

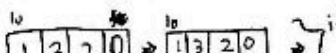

p i , EXECUTE "swap (x[lo], x[p])"

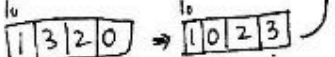


EXECUTE FIRST!
qsort(x, lo, p-1)
$i=0$ (0), $j=3$ (1), $k = \lceil \frac{3}{2} \rceil = 2$ (2)

EXECUTE SECOND
qsort(x, p+1, hi)
$i=5$ (8), $j=8$ (5), $k = \frac{\lceil 5+8 \rceil}{2} = 7$ (7)

We find 1 is median, swap w/ 1st value:

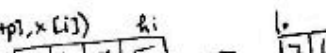
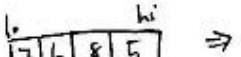

p i ↳ no swap p ↳ no swap

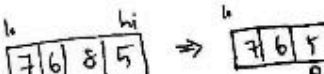
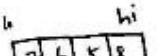

p ↳ swap i

EXECUTE "swap(x[lo], x[p])"

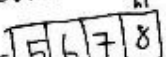
We find 7 is median, swap w/ first element:

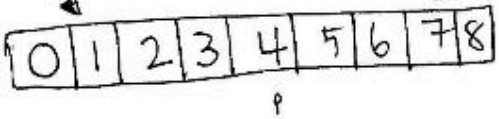
EXECUTE if(x[i] < x[lo]) swap(x[++p], x[i])


p i ↳ swap ⇒ 
p i ↳ no swap


p i ↳ swap ⇒ 
p i ↳ no swap

EXECUTE "swap(x[lo], x[p])"

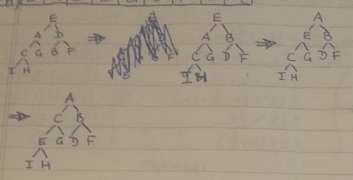



p

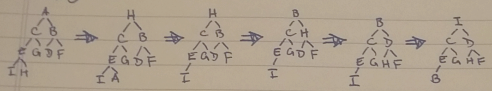
Mitchell Chan 47y9a
Sandra Yoo a9x9a

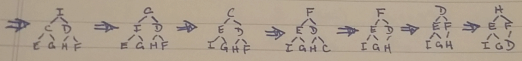
Date

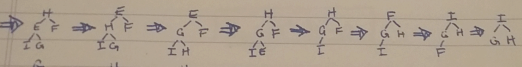
5a) | E | A | D | C | G | B | F | I | H |

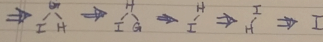


5b) Swap first and last node.
Delete last node from heap.
Heapify.









| A | B | C | D | E | F | G | H | I |

Mitchell Chan r7y9a
Sandra Yoo a9x9a

Date

6) Starting at the root of the heap, check left and right subtrees. For each subtree, if root value $< x$, then check left and right subtrees. It takes $O(k)$ time for this algorithm because it's impossible for there to be a key $> x$ in T to have a descendent that has a key $< x$.

~~7) Let the loop invariant be $I(n)$: xy + product = AB. The guard condition G of the while loop is G: $y < 0$ or $y \neq 0$. We will use the loop invariant theorem to prove that the while loop is correct with respect to the given pre and post conditions.~~

8)

i	a	pow
0	4	1
1	4	a
2	4	a*a
3	4	a*(a*a)
4	4	a* (a*a*a)

pre-condition: $i=1$, $pow=1$, $i \leq n$

~~post-condition~~

IH/Loop invariant

IH: $pow = a^n$

Base case: $n=0 \Rightarrow pow = a^0 = 1$

IS: Assume at the start of the k^{th} iteration, $pow = a^k$. At the start of the $(k+1)$ iteration, we will have $pow = pow * a$

$= a^k * a = a^{k+1}$

Termination: The loop terminates when $i=n$. When it ends, we are about to enter the $i=n+1$ iteration. By the loop invariant, when the loop ends, $pow = a^n$.

Page

#7.

Let the loop invariant be $\boxed{I(n): xy + \text{product} = AB}$

The guard condition G of the while loop is $\boxed{G: y!=0, \text{ or } y \neq 0}$

we will use the loop invariant theorem to prove that the while loop is correct with respect to the given pre and post conditions.

I. Basis property (I(0) is true before the 1st iteration of the loop)

I(0) is $xy + \text{product} = AB$, where $x \neq 0$ and $y \neq 0$ because by the precondition, $x = A$, $y = B$ and A and B must be positive integers. Also by the precondition, product = 0. Since $xy + \text{product} = AB$ from the precondition is true before the first iteration of the loop, I(0) is true.

II. Inductive property (If $G \wedge I(k)$ is true before a loop iteration (where $k \geq 0$), then $I(k+1)$ is true after the loop iteration)

Suppose k is a non-negative integer such that $G \wedge I(k)$ is true before an iteration of the loop. We will show that $I(k+1)$ is true after the loop iteration.

Since we know that G is true, $y_{old} \neq 0$ and we enter the loop. Before the execution of statement 1, since we know I(k) is true,

$$x_{old}\, y_{old} + \text{PRODUCT}_{old} = AB, \text{ where } x_{old} \neq 0 \text{ and } y_{old} \neq 0$$

The execution of statement 1 has the following effect:

$r_{new} = y_{old} \bmod 2$, and by the quotient remainder theorem,

$y_{old} = 2q + r_{new}$, where $q \in \mathbb{Z}$, $0 \leq r_{new} < 2$, $2q = k$

If $r_{new} = 0$, then after the execution of the 1st if statement,

$$x_{new} = 2x_{old} \quad \Rightarrow x_{new} y_{new} = \left(2x_{old}\right)\left(\frac{y_{old}}{2}\right)$$

$$y_{new} = \frac{y_{old}}{2}$$

$$= x_{old}\, y_{old}$$

$$\text{Product}_{new} = \text{product}_{old}$$

$$x_{new} y_{new} + \text{product}_{new} = x_{old}\, y_{old} + \text{product}_{old}$$

$$= AB$$

If $r_{new} = 1$, knowing that $y_{old} = 2q+1$, after execution of the first statement in the 2nd if statement,

MITCHELL CHAN R7Y9A
SANDRA YOO A9X9A

$x_{new} = x_{old}$

$Product_{new} = Product_{old} + x_{old}$, and after execution of the 2nd statement in the 2nd if statement,

$y_{new} = y_{old} - 1$

$y_{new} = (2q+1) - 1$ (substitution from above)

$y_{new} = 2q$ (Note: in our case, $2q$ is k)

We now get

$x_{new} \cdot y_{new} + Product_{new} = x_{old} 2q + product_{old} + x_{old} = x_{old} 2q + x_{old} + product_{old}$

We factor out the x_{old} and we get

$= x_{old}(2q+1) + product_{old}$ $(x_{old}(k+2) + product_{old})$

$= x_{old} y_{old} + product_{old}$

$= AB$

Hence, I(k+1) is true.

III. Eventual falsity of guard (after a finite number of iterations of the loop, G becomes false)

The guard G is the condition $y \neq 0$. By I and II it is known that for all $y > 0$, the loop is iterated then $xy + product = AB$. Each value of y obtained by repeated iteration will reach a least value, y_{min}. When $y_{min} = 0$, the guard G is false.

IV. Correctness of the post-condition (If N is the least number of iterations after which G is false and I(N) is true, then the values of the algorithm variables will be as specified in the post condition)

Suppose that for some non-negative integer N, G is false and I(N) is true.

Since G is false, $y = 0$, and since I(N) is true,

$x(0) + product = A \times B$

$0 + product = A \times B \Rightarrow product = AB$

RESOURCES USED: Discrete Mathematics w/ Applications by Susanna Epp (used to follow the structure of loop invariant theorem, used similar structure and style and wording to examples on P. 282-287)

(#11) Theorem: A ternary tree of height h has at most $\frac{3^{h+1}-1}{2}$ nodes.

MITCHELL CHAN R7Y9A
SANDRA YOO A9X9A

PROOF

We will use proof by induction.

We know if the height of the ternary tree is h, we can say the maximum number of nodes in a ternary tree is

$$3^0 + 3^1 + 3^2 + \dots + 3^h = \sum_{i=0}^{h} 3^i = \frac{3^{h+1}-1}{2}$$

Base case: when $h = -1$, it is an empty tree (as stated in the question)

when $h = 0$, we get

$3^0 = 1$ for the LHS, and

$\frac{3^{0+1}-1}{2} = 1$ for the RHS. Since LHS=RHS, the theorem holds for $h = 0$.

Induction hypothesis

we assume that the theorem holds for $h=K$,

$3^0+3^1+3^2+\dots 3^K = \frac{3^{K+1}-1}{2}$, so let us prove this for $h=K+1$.

If we add 3^{K+1}, by the induction hypothesis we get

$$3^0+3^1+3^2\dots 3^K+3^{K+1} = \frac{3^{K+1}-1}{2} + 3^{K+1}$$

$$= \frac{3^{K+1}-1+2\cdot 3^{K+1}}{2} = \frac{q^{K+1}-1}{2} = \frac{3\cdot 3^{K+1}-1}{2}$$

$$= \frac{3^{(K+1)+1}-1}{2}$$

Thus, the theorem holds for $h=K+1$, and we are done. QED.