M. SATHISH KUMAR
E0119052

DESIGN & ANALYSIS OF ALGORITHMS

FINAL ASSESSMENT

1. a) $O(\log n)$ → binary Search

Search a sorted array by repeatedly dividing the search interval in half. If the value of search key is less than the item, narrow the interval to the lower half. Otherwise, to the upper half.

b) $O(n)$ → Displaying the names of all employees

c) $O(n)$ → First name of an employee when jumbled

d) $O(1)$ → Displaying salary of employee at i^{th} index

2* The stack S contains size n. The exhaustive trials involves ~~cost~~ worst-case of $O(n)$ operations. Then for n operations, it will take $O(n^2)$.

Then each push and pop is done at most once.

* For empty stack push, pop and multipop takes almost $O(n)$ time

Amortized cost (Average Cost) $= \frac{\text{Worst Case}}{n} = \frac{O(n)}{n} = O(1)$

(i) Push (S,x) pushes the element into stack for ammortized O(1)

(ii) pop (S) pops the top of stack S returns the popped element since each of operation runs O(1) times

(iii) Multipop (S, 4) and Multi-Pop (S, 7)

for n values it multipops n time O(n)

It does one by one So amortized is O(1)

---

3. In hill cipher, the whole plaintext is divided into column vector of size 2×2 whereas in playfair cipher, we need to find diagraph for the given plaintext & 5×5 key matrix

Therefore in this case, the plaintext "Google" to cipher the text, hill cipher has more advantages like matrix multiplication, finding frequency.

So here hill cipher is advantageus as it can reduce letter redundancy and increase in performance speed

In hillcipher, GOOGLE is divided as

$$\begin{bmatrix} G \\ O \end{bmatrix} \begin{bmatrix} O \\ G \end{bmatrix} \begin{bmatrix} L \\ E \end{bmatrix}$$

'GO', 'OG', 'LE' are the diagraphs.

4\. Here we can use point polygon method.

→ According to this method, we have to extend a ray in any direction ~~f~~ from the point given.

→ If the extended ray intersects the ~~pt~~ polygon odd number of times, then it is within the polygon, if it ~~inte~~ intersects even number of times then it is outside the polygon.

→ Hence according to the designed algo, P_1 & P_2 are within the polygon as it intersects only one time (odd)

→ P_3 & P_4 intersects the ~~pt~~ polygon 2 times (even), hence they are outside the polygon

---

5\. The above described process is bully algorithm used for detecting the next leader

a) n - 1 process totally begin election as each every process will send message to its high priority process

b) $O(n^2)$ as there are n processes & (n-1) process begin election, there n (n-1) messages.

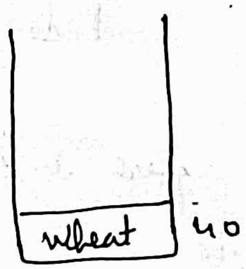
Hence the assymptotic complexity is $O(n^2)$ for message overhead.

7. $M = 60$ Kg

S. No	Items	Weight	Price	Price / weight ratio
1.	Rice	12	120	10
2	Wheat	20	240	12
3	Barka	15	150	10
4	Millet	16	64	4
5.	Cereals	5	50	10

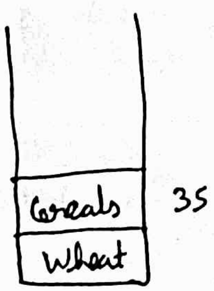
~~...~~ Greedy Techique:

- Step 1: Insert the item with the highest P/W ratio (i.e) wheat



$60 - 20 = 40$

- Step 2: Insert the second hight p/w ratio (cereals)



$40 - 5 = 35$

$60 - 20 = 40$

Step 3 :

Insert the third highest P/W ratio and subtract 12 from the remaining weight

Stack	
Rice	$35 - 12 = 23$
Cereals	$40 - 5 = 35$
wheat	$60 - 20 = 40$

Step 4:

Insert the fourth highest P/W ratio item i.e Bajra and subtract 15

Stack	
Bajra	$23 - 15 = 8$
Rice	$35 - 12 = 23$
cereals	$40 - 5 = 35$
wheat	$60 - 20 = 40$

Step 5:

Insert the fifth highest P/W ratio item is millet and ~~subt~~ subtract from remaining weight

Stack	
millet	$8 - 8 = 0$
Bajra	$23 - 15 = 8$
Rice	$35 - 12 = 23$
cereal	$40 - 5 = 35$
wheat	$60 - 20 = 40$

profit = p_i

32
150
120
50
240

592 → Total profit by fractional knapscale

Asymptotic performance Knapscale → $\alpha(n\log n)$

→ Step 6 → total profit gained is 592

→ Assymptotic performance of Knapscale with greedy technique is } $O(n)$

* So 0/1 Knapscale is helpful the find the local maximum profit value.

---

8. $\alpha = 17$, $\beta = 11$ (prime numbers)

RSA algorithm :

* Take 2 prime numbers α and β and find

$n = \alpha . \beta$

* Also find $\phi(n) = (\alpha - 1)(\beta - 1)$

* Find the value of e (public key) such that e is co-prime with $\phi(n)$ $\gcd(e, \phi(n)) = 1$

* Find the private key d which is less than $\phi(n)$ such that $de = 1 \bmod \phi(n)$.

* The generated public key pair is ~~PU{~~ PU{e, n} and private key pair is PR{d, n}

* Find the cipher text $C = M^e \bmod n$ (M - message value)

* For decryption, $M = C^d \bmod n$ (using exponential of private key acquired)

- Now using the above algorithm:

Step 1 : $n = 17 \times 11 = 187$

Step 2 : $\phi(n) = 16 \times 10 = 160$

Step 3 : $\gcd(e, 160)$ must be 1

Let us chose $e = 7$ as 17 and 160 are coprime

Step 4 : $d \times 7 = 1 \bmod 160$ Then $d = \frac{1}{7} \bmod 160$

Then multiplicative inverse of 7 under modulo 160 is 23

$d = 23$

PU{7, 187} and PR{23, 187}

- Step 5: Encryption

Given $m = 88 \Rightarrow C = 88^7 \bmod 187$

$88^2 \bmod 187 = 77$

$C = [(88^2 \bmod 187)(88 \bmod^2 \bmod 187)$
$(88^2 \bmod 187)(88 \bmod 187)] \bmod 187$

$= (66 \times 88) \bmod 187$

$= 11$

- Step 6: Decryption

$m = c^d \bmod n = 11^{23} \bmod 187$

$m = 11^{2+4+3+1} \bmod 187$

$11 \bmod 187 = 11$ $11^4 \bmod 187 = 55$

$11^2 \bmod 187 = 121$ $11^8 \bmod 187 = 33$

$m = (11 \times 55 \times 121 \times 33) \bmod 187$

$m = 88$

RSA provides ~~def~~ digital signature as it signs messages. It uses exponentiation to power d as e using private and public key.

So the reciever can get the encrypted value to power of e. If both values are ~~sam~~ same, the reciever is assured that message is not changed. Hence authentication cant be forged easily

10. Execution time = 110 seconds

Sequential cost => 25 seconds (C_s)

Parallelization time = 110 - 25 = 85 (C_p)

(i) $S(p, n)$ for $n = 1$

$$T_1 = C_s + C_p n^2 = 25 + 85n^2$$

$$T(p,n) = C_s + \frac{C_p n^2}{p} = 25 + \frac{85 n^2}{p}$$

$$S(p,n) = \frac{25 + 85n^2}{25 + \frac{85 n^2}{p}}$$

$P = 5, n = 1$

~~$S = C_p$~~

$$S_{(p,n)} = \frac{25 + 85(1)^2}{25 + \frac{85(1^2)}{5}}$$

$$= \frac{110}{42} = 2.619 \simeq 2.62$$

(11) Case I :

$n=4, p=5$

$$S(4,5) = \frac{25+85(16)}{25+\frac{17}{\cancel{85}}(16)}{\cancel{5}}$$

$$= \frac{1385}{297} = 4.66$$

$S(4,5) = 4.66$

$$\text{Execution time} = C_s + \frac{C_p n^2}{p}$$

$$= 25 + \frac{85}{5}(4^2)$$

$$= 297 \text{ seconds}$$

Case II :

$n=4, p=15$

$$\text{Execution time} = \frac{25+\overset{17}{\cancel{85}}(42)}{\cancel{15}_{3}}$$

$$= 90.67 + 25$$

$$= 115.67 \text{ seconds}$$

$$S(4,15) = \frac{25+85(4^2)}{115.67}$$

$$= \frac{1385}{115.67} = 11.97$$

(ii) 25 % parallelized ~~&~~ cost

$\Rightarrow$ 15 % serial cost

$\therefore f = 0.15$

$P = 15$

$$S(p,n) = \frac{1}{f + \frac{1-f}{p}}$$

$$S(15,n) = \frac{1}{0.15 + \frac{0.85}{15}}$$

$$= \frac{1}{0.206} = 4.854$$

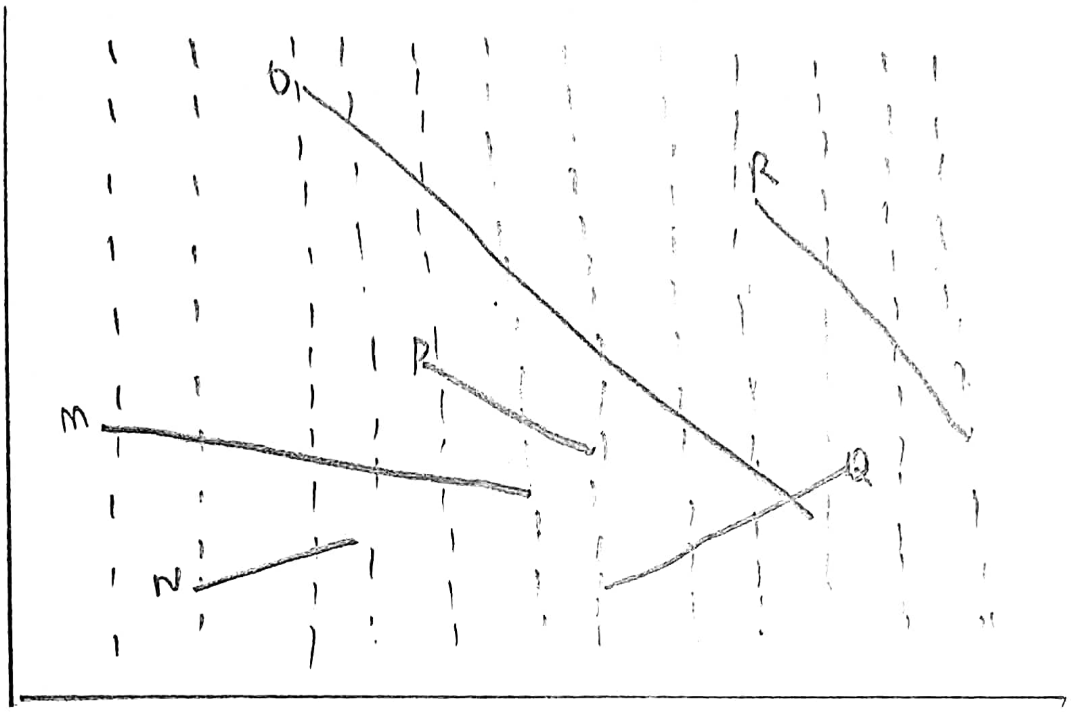
~~P=15~~ $S(g,n) = 10$

$$10 = \frac{1}{f + \frac{1-f}{15}} = \frac{15}{f(15-1)+1}$$

$$2\ \cancel{10} = \frac{\cancel{15}\ 3}{14f+1} = 28f + 2 = 3$$

$$\boxed{f = 0.36}$$

9. a) There are 6 line segments N, M, P, O, Q, B and R. The dotted lines are the sweep line when moves vertically it shows like this.



Sweep lines

Here all points from left to right are processed one by one. We maintain a self balancing binary search tree

* left end point of line segment M is processed: M is inserted into the tree. The tree contains M.
  Here No intersection

* Left end point of line segment N is processed:
  Intersection of M and N is checked. N is inserted into tree

* Left End point of line segment O is processed:
  Intersection of O with M is checked.
  No interaction. O is inserted to tree.

* Right end point line segment N is processed:
N is deleted from tree. Intersection of M and O is checked

* Left end point of linesegment P is processed:
Intersection of P is checked with M and O. No intersection. Tree contains M, O, P

* Left end point of line segment Q is processed:
Q is added to the tree and checked with M, P, O. Intersection occurs with Q and O.

* Right end point of line segment P is processed:
P is deleted from tree. No intersection. The tree contains: M, O.

* Right end point of O is processed:
O is deleted from tree

* Left end point of R is processed

* Right end points of R and Q are processed
Both are deleted from tree and tree becomes empty

E0119052 (14)

★ Time complexity:

The first step is sorting which takes O(nlogn) times. The second step is n points and for processing every point, it takes O(logn) time.

Overall time complexity: O(nlogn)

b) <u>Circle event</u>:

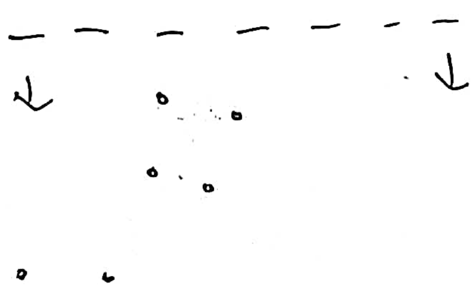
. When the ⊥ of all sites gets intersects forming a voronoi ~~vertex~~, circular events takes place.

<u>Time</u> <u>complexity</u>:

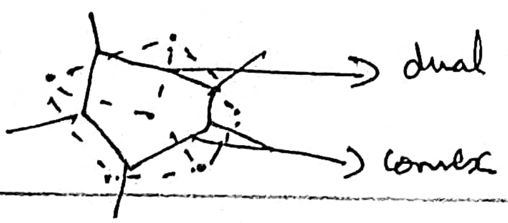
Naive approach → $O(n^4)$

Instrumental " → $O(n^2)$

Baseline

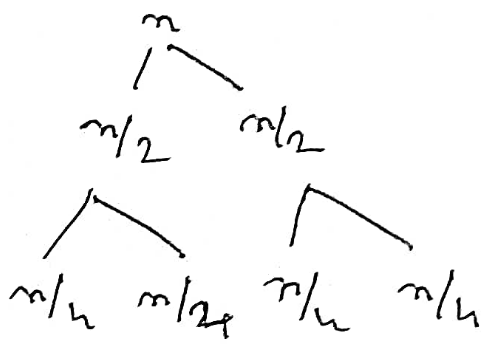


The final voronoi diagram for above point;



6. Sorting algorithm of any method is possible to find max and min profit in a month. The efficient one is Quick sort with $T(n) = O(n\log n)$

$$T(n) = 2T(n/2) + n$$



Level 0 : $n/2^0$ Level i = $\frac{n}{2^i}$

Level 1 : $\frac{n}{2^1}$

$\frac{n}{2^n} = < 1$ at last level

cost of last level

$= 2^{\log_2 n} = O(n)$

$$T(n) = \{\underbrace{n + n + \dots}_{\log n \text{ levels}}\} + \Theta(n)$$

$$= n\log n + \Theta(n) = \Theta(n\log n)$$