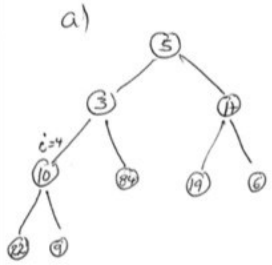
AMANUEL EMHA

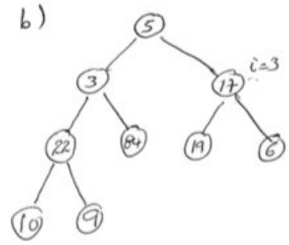
Question 1 (a)

$$i = \left\lfloor \text{length}[A]/2 \right\rfloor = \lfloor 9/2 \rfloor = 4$$

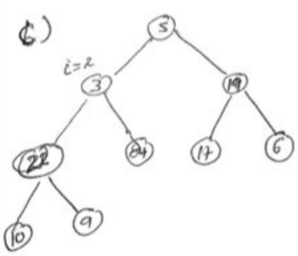
a)



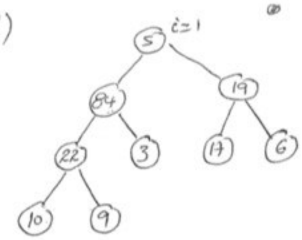
b)



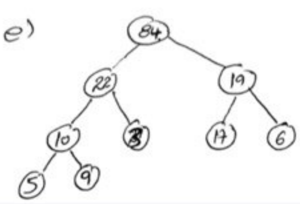
c)



d)



e)



QUESTION 1

(b) Red black tree with n-internal nodes has height at most $2\lg(n+1)$

PROOF BY INDUCTION

1- BASE CASE

Left Side : Since $x=0$ $bh(x)=0$

Right Side : If $bh(x)=0$ then $2^{bh(x)}-1 = 2^0-1 = 1-1 = 0$

2- BY Inductive

Total number of nodes at height $bh(x) \geq 2^{bh(x)-1}-1+2^{bh(x)-1}-1+1$

$\Rightarrow 2^{bh(x)} \cdot 2^{-1} - 1 + 2^{bh(x)} \cdot 2^{-1} \cancel{-1} \cancel{+1}$

$\Rightarrow 2^{bh(x)} - 1$ Since $bh(x) \geq h/2$

$\Rightarrow n \geq 2^{h/2} - 1$

$\Rightarrow 2^{h/2} \leq n+1$ take log on both sides

$\Rightarrow h/2 \leq \lg(n+1)$

$\Rightarrow h \leq 2\lg(n+1)$

Question 1

(c) Successor of 13 is 15

Successor of 6 is 7

Predecessor of 6 is 4

Question 2 a - Solution

To find $B[i, w] = \max\{B[i-1; w]; w_i + B[i-1, w-w_i]\}$

The table Consisting of $B[i, w]$ Values is:

i \ w	1	2	3	4	5	6	7	8	SELECTED ITEMS
2	0	3	3	3	3	3	3	3	2
3	0	3	4	4	7	7	7	7	2,3
4	0	3	4	5	7	8	9	9	3,4
5	0	3	4	5	7	8	9	10	3,5

- If we see the table, ~~lets~~ say third line it is the result we get when we were trying to find the maximum benefit using the set of items {2, 3, 4}.

- Maximum benefit, over the set of all items is **10** and {3,5} are the set of items that gives us the benefit (10). (based on what we found on the table)

- Tracing backwards through the table

  Example $B = 10 - 6 = 4$

  weight $= 8 - 5 = 3$

9:36 LTE

2021-08-12 10-39 1.pdf

Question 2b)

b) Running time of 0-1 knapsack using #a above is $O(nW)$ where n is the number of items and W is the limit on the weight.

$Size = \lg_b W \Rightarrow W = 2^{size}$ where Size = n° of bits to represent 'W'.

b = base

$\therefore T(n) = O(n.W) \Rightarrow O(n \times 2^{size})$

$\Rightarrow O(nW)$

Question 3 (a)

DIJKSTRA (G, w, s)

1- INITIALIZE-SINGLE-SOURCE ——— $O(V)$
2- $S = \emptyset$ ——— $O(1)$
3- $Q = G.V$ ——— $O(V)$
4- while $Q \neq 0$ ——— $O(V$
5- U = EXTRACT MIN (Q) ——— $O(V \lg V)$
6- $S = S \cup \{u\}$ ——— $O(V \lg V)$
7- foreach Vertex $V \in G.Adj[u]$ ——— $V(\bar{\Delta E} \cdot \lg V) = V \cdot \bar{\Delta E} \sim 2E$ i.e $O(2E) = O(E)$
8- Relax ——— $O(E \lg V)$

Every time we enter line 7 V times we relax only neighboring nodes i.e we call it $\bar{\Delta E}'$ (Delta E')

$\therefore$ Total we can say that this are

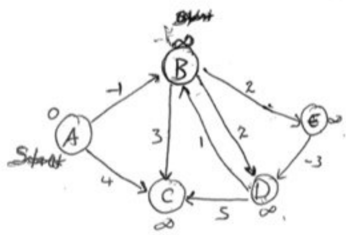
$V \cdot \bar{\Delta E}' \sim 2E \sim$ i.e $O(2E) = O(E)$

So the total running time is $\left[\begin{array}{l} O(V \lg V) + E \lg V \\ = O[(V+E) \lg V] \\ = \underline{O(E \lg V)} \end{array}\right.$ we know $E \gg V$

~~Bellm~~ Question (3b)

Bellman's-for Question (3b)

(B,E), (D,B) (B,D) / (A,B) (A,C) / (D,C) (B,C) (E,D)



1st Iteration

A	B	C	D	E
0	-1	∞	∞	∞
0	~~4~~	4	∞	∞
0	-1	2	∞	∞

- relax
- min (a)

When all edges are Processed first time.

2nd Iteration

A	B	C	D	E
0	-1	2	∞	1
0	-1	2	1	1
0	-1	2	-2	1

- relax
- min (a)

Result after 2nd Iteration

Third & fourth Iteraction - Same

A	B	C	D	E
0	-1	2	-2	1

- relax
- min (a)

So the result is Distance from A to All other Nodes

	Distance	Prev. Vertex
A	0	0
B	-1	A
C	2	B
D	-2	E
E	1	B

Question 3C

UNIVERSAL HASHING

- Universal hashing is Hashing function $\frac{|H|}{m}$.
- It is used to minimize Collision and the probability of Collision will be $1/m$ using Hash function.
- To prove this $h(x) = h(y)$

$$\sum_{i=0}^{r} a_i x_i \bmod m = \sum_{i=0}^{r} a_i y_i \bmod m \implies \sum_{i=0}^{r} a_i x_i \bmod m - \sum_{i=0}^{r} a_i y_i = 0$$

$$a_0 (x_0 - y_0) + \sum_{i=0}^{r} a_i (x_i - y_i) \bmod m = 0$$

$$a_0 (x_0 - y_0) = -\sum_{i=0}^{r} a_i (x_i - y_i) \bmod m$$

$$a_0 = \sum_{i=1}^{r} a_i (x_i - y_i) (x_0 - y_0)^{-1} \bmod m$$

The non-zero quantity $(x_0 - y_0)$ has a multiplicative inverse modulo m and there is a Unique Solution for a_0.

- For fixed Value $a_1, a_2 \dots a_r$ there is exactly one Value of a_0 that Satisfies $h(x) = h(y)$
- Now there are m^{r+1} hash functions, and m^r Possible Collisions

Probability of Collision $= \frac{m^r}{m^{r+1}} = 1/m$

- No it is impossible to have Collision-less hash since it has fixed length.

~~Subset Sum~~

Question 5 (a)

Answer

SubsetSum $\xrightarrow[\text{Poly}]{\text{reduce}}$ Knapsack $\Rightarrow$ SubsetSum $\leq_p$ knapsack

Show Knapsack is NP-C problem. To Show that:

1^{st} - Show that Knapsack belongs to NP.

Let's say we have an input set, ~~always~~ check if the total weight is at most W and if the corresponding profit is at least V. It takes only linear time to add all profits and weights to find true/false result of the decision problem. Thus we can verify solution in poly time.

2^{nd} Use SubsetSum Problem to reduce it to Knapsack Problem

Instance: Non-negative weights $w_1, w_2, w_3 \cdots w_n, W$
Profits $V_1, V_2, V_3 \cdots V_n, V$

Question: Is there a subset of weights with total weight at most W, such that the corresponding profit is at least V?

SubSetSum Problem

Instance: Non-negative integer numbers $s_1, s_2, s_3 \cdots s_n$ and t

Question: Is there a subset of these numbers with a total sum t?

To reduce an instance of SubsetSum to an instance of Knapsack Prob. we can create a knapsack that has the following:

$w_i = c_i = s_i$ $\qquad W = V = t$