CSCI-5454-Algorithms

1] a) $f(n) = \sum_{j=1}^{n} j^{10}$

By definition of Big-O

$f(n) \in O(n^{11})$ if there exists a constant c & n_0 such that for all $N \geq n_0$ we have $f(n) \leq c \cdot n^{11}$

$$\sum_{j=1}^{n} j^{10} \leq c \cdot n^{11}$$

$$c \geq \frac{\sum_{j=1}^{n} j^{10}}{n^{11}}$$

$$c \geq \frac{1^{10} + 2^{10} + \dots + n^{10}}{n^{11}}$$

$$c \geq \frac{(n-(n-1))^{10} + (n-(n-2))^{10} + \dots + n^{10}}{n^{11}}$$

As we're looking for worst case performance we can replace all terms by n^{10} as $[n-(n-1)] \leq n^{10}$ & $[n - \frac{n}{2}] \leq n^{10}$.

$$\therefore c \geq \frac{n^{10} + n^{10} + \dots n^{10}}{n^{11}} \geq \frac{n^{11}}{n^{11}} \geq 1$$

$\therefore c \geq 1$

For $c \geq 1$ & $n_0 \geq 1$ $f(n) \leq c \cdot n^{11}$

Hence $f(n) \in O(n^{11})$

b) $f(n) = \sum_{j=1}^{n} j^{10}$

By definition of Big Ω

$f(n) \in \Omega(n^{11})$ if $f(n)$ can be expressed as
$f(n) \geq C \cdot n^{11}$ where $n_0 > 0$ & there exists a $c > 0$.

$$\sum_{j=\frac{n}{2}}^{n} j^{10} \geq C \cdot n^{11}$$

ie $$c \leq \frac{\left(\frac{n}{2}\right)^{10} + \left(\frac{n}{2}+1\right)^{10} + \cdots + \left(\frac{n}{2}+\frac{n}{2}\right)^{10}}{n^{11}}$$

neglicting constants & replacing all terms with just $\frac{n}{2}$ as we're checking for lower bounds

$$c \leq \frac{\left(\frac{n}{2}\right)^{10} + \left(\frac{n}{2}\right)^{10} + \cdots + \left(\frac{n}{2}\right)^{10}}{n^{11}}$$

$$c \leq \frac{\left(\frac{n}{2}\right)\left(\frac{n}{2}\right)^{10}}{n^{11}} \leq \frac{n^{11}}{2^{11}\, n^{11}} \leq \frac{1}{2^{11}}$$

For $c \leq \frac{1}{2^{11}}$ we can say $f(n) \geq C n^{11}$

$\therefore f(n) \in \Omega(n^{11})$

1.c] By definition of o

$$\text{if } \lim_{n \to \infty} \frac{f(n)}{g(n)} = 0 \text{ then } f(n) \in o(g(n))$$

$$g(n) = n^{12}, \quad f(n) = \sum_{j=1}^{n} j^{10}$$

$$\therefore \lim_{n \to \infty} \frac{\sum_{j=1}^{n} j^{10}}{n^{12}} = \frac{1^{10} + 2^{10} + \dots + n^{10}}{n^{12}}$$

We know $1^{0} \leq n^{10}$, $2^{10} \leq n^{10}$ ... so neglecting lower order terms & replacing them with upper bound n^{10}

$$\lim_{n \to \infty} \frac{n^{10} + n^{10} + \dots + n^{10}}{n^{12}} = \lim_{n \to \infty} \frac{n^{11}}{n^{12}} = \lim_{n \to \infty} \frac{1}{n}$$

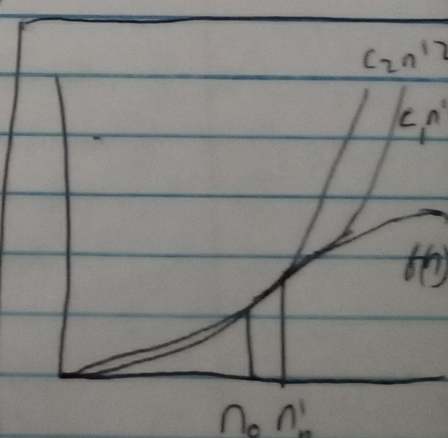
$$= \frac{1}{\infty} = \frac{1}{\frac{1}{0}} = 0$$

$$\therefore \lim_{n \to \infty} \frac{f(n)}{g(n)} = 0.$$

$$\therefore f(n) \in o(g(n)) = o(n^{12})$$

OR.

From 1a we know that $f(n) \in O(n^{11})$
$f(n) \leq c_1 n^{11}$ for some $c_1 > 0$ & $n_0 \geq n_0$.
we also that
$n^{11} \leq c_3 n^{12}$ if $n^{11} \in O(n^{12})$.
If a function $f(n)$ is upper bounded by $O(n^{11})$ it also means that it can be loosely bound by $o(n^{12})$.
Hence $f(n) \in o(n^{12})$.



By definition of Θ given $f(n)$ & $g(n)$
if we can find c_1 & c_2 such that

$c_1 g(n) \leq f(n) \leq c_2 g(n)$

then $f(n) \in \Theta g(n)$
from a & b $c_1 = \frac{1}{2^{11}}$ & $c_2 = 1$

Hence we can say.

$\frac{1}{2^{11}} n^{11} \leq f(n) \leq 1 \cdot n^{11}$

$\therefore f(n) \in \Theta(n^{11})$

3.a) Path is a sequence of vertices $v_1, \dots, v_k$ and for each vertex (v_i, v_{i+1}) in path P where v_0 & v_{i+1} are not v_1 & v_k there is an edge $e(v_i, v_{i+1}) \in E$ that connects two vertices.

Cycle is a path where first & last vertices are same. $v_1, \dots, v_k$ is a path which forms a cycle when $v_1 = v_k$ &

1)

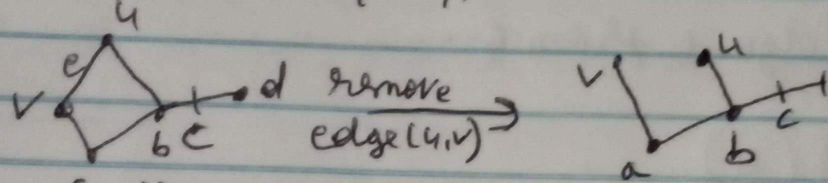
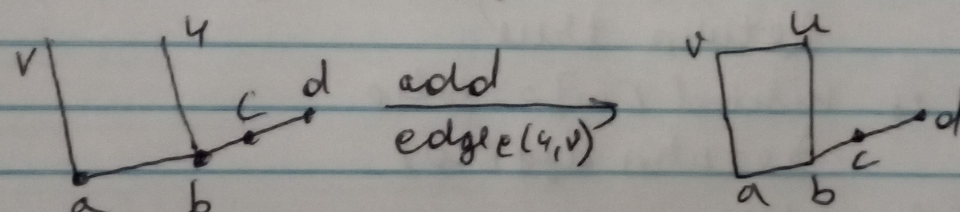


fig 1 fig 2

Graph G has a cycle with vertices u, v, a, b, u when we remove edge (u, v) we get a path u, b, a, v. Vertex v in fig 2 is still reachable after edge e is removed via other n-2 vertices which formed cycle in figure 1. v is reachable through a & b.

2]



Graph G has path from u to v through b, a which is u→b→a→v. If we add an edge u, v to the graph we form a cycle with path as u, b, a, v, u.

So from ① & ②. If we remove e from G there is still path to u, v. If they form a cycle. If there is a path from u to v & we add edge e(u,v) it forms a cycle. Hence, G has a cycle containing e if and only if when we remove e from graph, there exists path from u to v.

3b. Input: Graph $G = (V, E)$
Edge $e = (u, v)$

output: Return whether G has a cycle containing edge e.

For all vertices in V $O(V)$
    Initialize visited [vertex] = false

Remove edge e from G.

call dfs(u)

```
dfs (x)            → called V times since O(V)
   mark visited [x] = true       O(1)
   For all neighbours w of x     O(E)
       if w == v
           visited[w] = true
           return true
       if visited [w] == false
           return dfs(w)
   return false.
```

__Correctness:__
we call dfs on vertex u of edge $e=(u, v)$.
If there exists a cycle in graph G such that edge (u, v) is part of the cycle then even when we delete edge $e(u, v)$ there exists a path from u, v thro other vertices which formed cycle before e was removed.

Running time:

Initialization step takes $O(v)$.
we call dfs on every vertex till we reach v
that is $O(V)$.
we go through neighbours of vertex which
it is connected to & not other vertices.
That would be $O(E)$ where $E \le 2V+1$

$\therefore$ Runtime doesn't exceed $O(|V|+|E|)$.

2b] We know that stable matching has a worst run time of $O(n^2)$. In this problem we have to come up with such a preference list that minimum number of iterations to get stable matches is $\Omega(n^2)$.

Proof: Consider preference list of 'n' men & women.

The worst case can be made best case by having each man propose each ~~to~~ each woman in the preference list before he finds his match. Each ends up making $(n-1)$ proposals before he matches stably with last woman.

Each ~~doctor~~ man has to repeat this n times and 1 where he matches so we have $n(n-1)+1$. worst case iterations.

This can be acheived where men have same preference list & women have their preference list in such a way that each man ends up making $(n-1)$ proposals.

$m_1 (w_1, w_2, \ldots w_n)$ $\qquad$ $w_1 (m_n, m_{n-1}, \ldots m_1)$

$m_2 (w_1, w_2, \ldots w_n)$ $\qquad$ $w_2 (m_{n-1}, \ldots m_1$.

$\vdots$ $\qquad$ $\vdots$

$m_n (w_1, w_2, \ldots w_n)$

$\qquad$ $w_n (m_1, \ldots)$

i) m_1 proposes to w_1 and forms a pair (m_1, w_1)

ii) m_2 proposes w_1 & m_1 becomes free & m_2 is ranked higher.

iii) m_1 then moves onto proposing every woman till he matches with w_n.

Similar to m_1 other men will be rejected & hence free & will propose to $(n-1)$ women before getting matched.

$\therefore$ number of iteration $= n(n-1)+1$

$= n^2 - n + 1$.

$\therefore$ Given above preference list algorithm takes minimum $\Omega(n^2)$ iterations.

For example:

$n = 2$

m_1	w_1	w_2
m_2	w_1	w_2

w_1	m_2	m_1
w_2	m_1	m_2

i) m_1 proposes w_1 as w_1 is free (m_1, w_1) form a pair.

ii) m_2 proposes w_1 and m_2 is ranked higher than m_1 on w_1's preference list. so m_1 is rejected & (m_2, w_1) form a stable matching.

iii) m_1 then proposes w_2 to form a stable match (m_1, w_2).

$\therefore$ No of iterations $= 3$

for $n = 2$ $\quad n^2 - n + 1 = 4 - 2 + 1 = 3.$

Hence proved.

References: Textbook & Class notes

People collaborated with: Nischal & Madhusudhan