Assignment-1

① Given: Algorithm A

Running Time $O(2^n)$

So, if $n=5$, running time $= 2^5 = 32$ instructions

Sunway TaihuLight — 10^{17} instructions/second.

problem:

1(a) (a) $n=100$

Time taken by sunway Taihulight to finish the algorithm with $n=100$.

1 second $\longrightarrow 10^{17}$ instructions

? $\longrightarrow 2^{100}$ instructions

$$x = \frac{2^{100}}{10^{17}} = (12,676,506,002,282 \cdot 2940 \ldots)$$

1 century $\longrightarrow$ 3,155,695,200 seconds

x centuries $\longrightarrow$ 12,676,506,002,282·2940 seconds

$$x = \frac{12,676,506,002,282}{3,155,695,200} = 4017 \text{ centuries}$$

So, Sunway Taihulight need 4017 centuries to finish the algorithm with 2^{100} instructions.

1(b) $n = 1000$

1 second $\longrightarrow$ 10^{17} instructions
x seconds $\longrightarrow$ 2^{1000} instructions

$$x = \frac{2^{1000}}{10^{17}} = 1.07150860\cdots 906e+284 \text{ seconds.}$$

1 century $\longrightarrow$ 3,155,695,200 seconds
x centuries $\longrightarrow$ 1.07150860 --- 906e+284 seconds

$$x = \frac{1.07150860\cdots 906e+284 \text{ seconds}}{3,155,695,200}$$

$$= \underline{3.395475 \cdots e+274} \text{ centuries}$$

So, Sunway Taihulight supercomputer needs 3.395475 ---e+274 centuries to finish the algorithm with 2^{1000} instructions.

② (a) n^2

Input size doubles $\Rightarrow (2n)^2$

$\Rightarrow \frac{4\cancel{n^2}}{\cancel{n^2}} = \underline{4 \text{ times}}$ slower.

(b) n^3

Input size doubles $\Rightarrow \frac{(2n)^3}{n^3} = \frac{8\cancel{n^3}}{\cancel{n^3}} =$

$= \underline{8 \text{ times slower}}$.

(c) $100n^2$

Input size doubles $\Rightarrow \frac{100(2n)^2}{100n^2} = \frac{4 \times \cancel{100n^2}}{\cancel{100n^2}}$

$= \underline{4 \text{ times slower}}$.

(d) $n \log n$

Input size doubles $\Rightarrow$ $\cancel{\frac{2n \log(2n)}{n \log n}}$ $\frac{2n(\log 2 + \log n)}{n \log n}$

$= \cancel{2 \log(2n-n)}$ $2\left(1 + \frac{1}{\log n}\right)$ times slower

$=$ ~~$2\log n$ times slower.~~

(e) 2^n

Input size doubles $\Rightarrow \frac{2^{2n}}{2^n}$

$= 2^{2n-n} = \underline{2^n}$ times slower.

③ ⓐ $f(n) = \cancel{100}n + \log n$ & $g(n) = \cancel{8}n + \log^2 n$

$$n + \log n \leq n + \log^2 n$$

$$\boxed{f(n) = \Theta\, g(n)}$$

$\cancel{n + \log n}$ $\cancel{f(n) \geq \Omega\, g(n)}$

ⓑ $f(n) = \cancel{20}\log n + \cancel{4}$ & $g(n) = \log n^2 - 100$

$$\log n = \cancel{2}\log n$$

$$\boxed{f(n) = \Theta\, g(n)}$$

ⓒ $f(n) = \dfrac{n^2}{\log n}$ and $g(n) = n\log^2 n$

$$\frac{n^2}{\log n} \geq n\log^2 n$$

$$\boxed{f(n) = \Omega\, g(n)}$$

ⓓ $f(n) = \sqrt{n}$ & $g(n) = \log^5 n$

$$\sqrt{n} \geq \log^5 n$$

$$\boxed{f(n) = \Omega\, g(n)}$$

(e) $f(n) = n2^n$ $\quad g(n) = 3^n$

$n2^n \leq 3^n$

$n \leq (1.5)^n$

$\boxed{f(n) = O\,g(n)}$

(f) $f(n) = 4n\log n$ $\quad g(n) = n\log_3 n$

$$4\,\frac{n\log n}{\cancel{\log 2}} = \frac{n\log n}{\cancel{\log 3}}$$

$\boxed{f(n) = \Theta(g(n))}$

Problem : 4

Here, In this algorithm the knapsack is taken as list. The ~~items of~~ size of the items are taken as a[i]. In starting i took list as empty and sum=0. Then the for loop starts.

If sum + a[i] ≤ K, then we added a[i] into the list. list.add(a[i]).
Next, we incremented the sum with size of a[i] which is k_i : sum = sum + k_i.
Next, check whether sum ≥ K/2. If it is satisfied, then break the algorithm. otherwise proceed with a[i+1]

else (sum + a[i] > K), then check whether a[i] is less than or equal to K. If this condition is satisfied add a[i] into the list and break the algorithm. otherwise proceed with a[i+1]

Here I wrote only one for loop from 1 to n where we considered each item only once, the running time of an algorithm is O(n).

Pseudocode :

```
size  = k1, k2, k3 ---- kn        } Input.
items = a[1], a[2] ---- a[n]      }

sum = 0 , list = ∅
for i = 1 to n
        if sum + ki ≤ K
            list.add(a[i])
            sum = sum + ki
            if sum ≥ K/2
                break;
        else
            if ki ≤ K
                list.add(a[i])
                break;
```

Output: knapsack list of factor 2 approx solution

Discussed with Shivakesh Reddy Annepally
Sahiti Katragadda