CSE 321 Homework 4

Ramazan KARAKAVAK
141044063

Q1) a) $A[i,j] + A[i+1,j+1] \leq A[i,j+1] + A[i+1,j]$

Let the $(i+1)$ = k → $A[i,j] + A[k,j] \leq A[i,j+1] + A[k,j]$ where $i < k$

induction method ⇒ assume $k = i+n$ ⇒ $k+1 = i+n+1$

↳ $A[i,j] + A[k,j+1] \leq A[i,j+1] + A[k,j]$

\+

↳ $A[k,j] + A[k+1,j+1] \leq A[k,j+1] + A[k+1,j]$

⇒ $A[i,j] + A[k,j+1] + A[k,j] + A[k+1,j+1] \leq A[i,j+1] + A[k,j] + A[k,j+1] + A[k+1,j]$

→ $A[i,j] + A[k+1,j+1] \leq A[i,j+1] + A[k+1,j]$

b) special array = 2D array (list of lists)

```
i=0
for r in specialArray
    j=0
    for c in r

        sumMustBeLess = specialArray[i][j] + specialArray[i+1][j+1]
        sumMustBeGreater = specialArray[i+1][j] + specialArray[i][j+1]
        if sumMustBeLess > sumMustBeGreater :

            specialArray[i][j] -= SumMustBeless - SumMustBeGreater
```

Calculate $A[i,j] + A[i+1,j+1] \leq A[i,j+1] + A[i+1,j]$ formula in the 2D array.

If left side greater than right side, change 1 element of this four elements.

c) Divide a row two ports.

Find minimum left port

Find minimum right port

compare this two

Continue..

do this for every row in matrix.

d) $T(m) = T(m/2) + cn + dm$

$= cn + dm + cn + dm/2 - -$

$= \sum_{i=0}^{\log m-1} cn + \sum_{i=0}^{\log m-1} \frac{dm}{2^i}$

$\Rightarrow cn \lg m + dm \sum_{i=0}^{\lg m-1} < cn \lg m + 2dm$

$\Rightarrow O(n \log m + m)$

2) Finding kth element of the merged array of these two sorted arrays.

- The best case is O(1) → if length of one of the array is 0, the answer is kth element of the second array. and the k = 1; best case is O(1)

- The worst case is k can be max (m+n) (length of arr1 + arr2)

⇒ logm + logn

3) Finding maximum subarray (contigous)

- first find maximum sub array left half
- find maximum subarray right half

find maximum subarray which crosses the midpoint and continue.

Best case if length of array == 1 ; $T(n) = \Theta(1)$ (the maximum sum is just one element)

Worst case ⇒ recurrence relation ⇒ $2T(n/2) + \Theta(n)$ ⇒ solve ⇒ $T(n) = \Theta(n \log n)$

4) Check bipartite of the graph

- Assign colour the source vertex
- Assign all the neighbours of the above vertex another colour.
- Taking one neighbour at a time, assign all neighbour's neighbours.
- continue
- If at any step, we find a neighbour which has been assigned the same colour as that of the current vertex, stop the process.

↳ the graph cannot be coloured using two colours. → Graph is not bipartite

I used adjacency list of my implementation. So Best case : O(V+E)

Worst case : O(V+E)

5) Finding gain

⇒ Best case → if the cost and price arrays have size 1, our best case is $O(1)$

⇒ Worst case ⇒ recurrence relation → $2T(n/2) + \Theta(n)$ ⇒ solve ⇒ $T(n) = \Theta(n \log n)$