Q 2.

$Y = 5$

$10 \cdot 5 = 50$

$11 \cdot 5 = 55$

$A[i] = [\underset{0}{50}, \underset{1}{55}, \underset{2}{50}, \underset{3}{53}, \underset{4}{51}]$

$B[i] = A[i] + 5$

$B[0] = A[0] + 5 = 50 + 5 = 55$

$B[1] = A[1] + 5 = 55 + 5 = 60$

$B[2] = A[2] + 5 = 50 + 5 = 55$

$B[3] = A[3] + 5 = 53 + 5 = 58$

$B[4] = A[4] + 5 = 51 + 5 = 56$

$B[i] = [\underset{0}{55}, \underset{1}{60}, \underset{2}{55}, \underset{3}{58}, \underset{4}{56}]$

Let $A[i]$ be the start time, $B[i]$ be the finish time

$A[i] = [\underset{0}{50}, \underset{1}{55}, \underset{2}{50}, \underset{3}{53}, \underset{4}{51}]$

activities : 1 2 3 4 5

$B[i] = [\underset{0}{55}, \underset{1}{60}, \underset{2}{55}, \underset{3}{58}, \underset{4}{56}]$

Activity 1 = 3 use activity = 1

$A[0] = A[2]$ start 50 -> 55 finish

$B[0] = B[2]$ skip activity = 3

next choose one of activities 2, 4, 5

$A[1] = 55 = B[0]$ (2) start 55 finish 60

$53 < 55$ 4 start 53 finish 58 skip

$51 < 55$ 5 start 51 finish 56 skip

So we use activity = 2

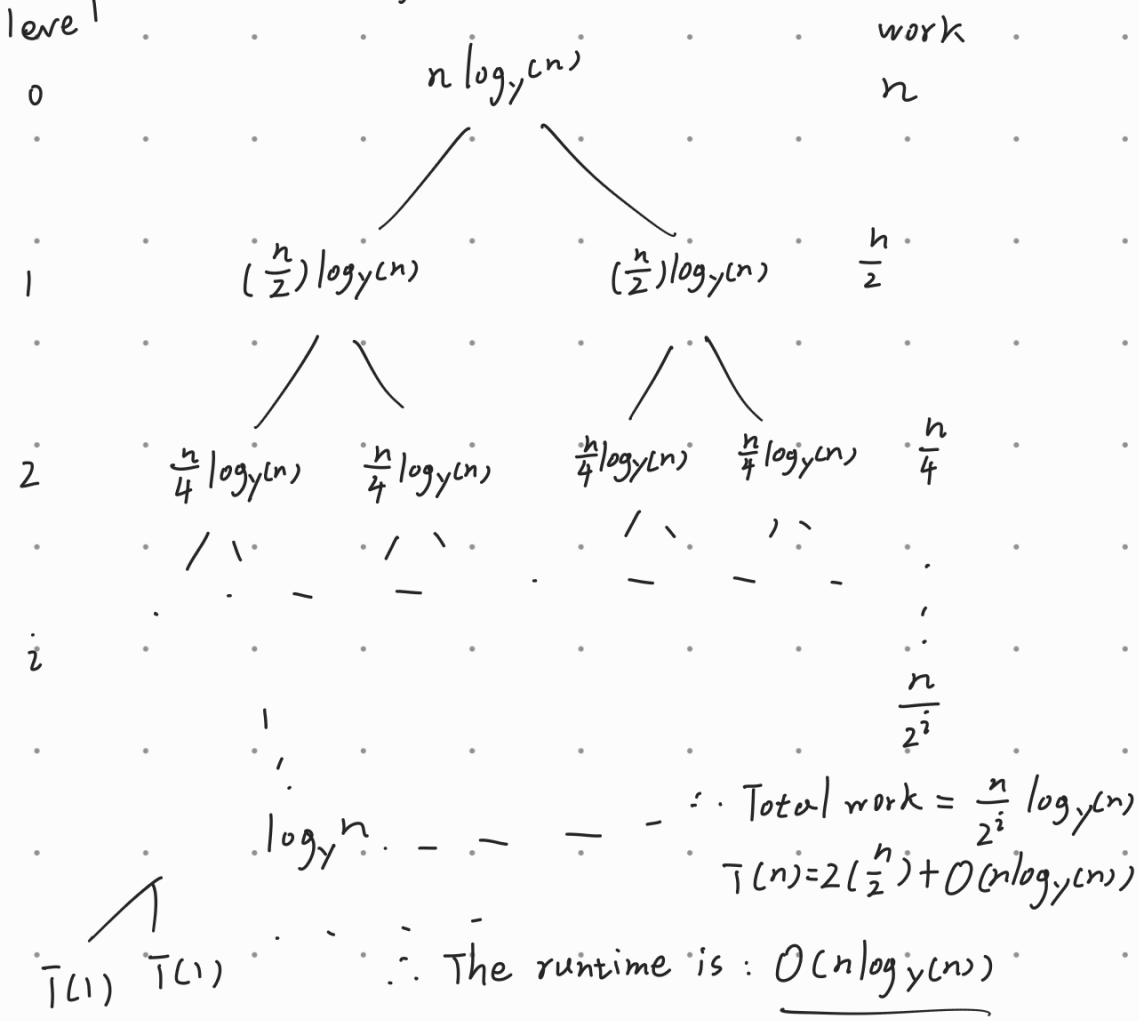
The greedy algorithm will selects activities 1 and 2 $A[0], B[0]$ and $A[1], B[1]$

or will selects activities 3 and 2 $A[2], B[2]$ and $A[1], B[1]$

Q3.

array with n strings, each string length: $\log_y(n)$

mergesort [...|...] $n\log_y(n)$



$\therefore$ Total work $= \frac{n}{2^i}\log_y(n)$

$T(n) = 2(\frac{n}{2}) + O(n\log_y(n))$

$\therefore$ The runtime is: $O(n\log_y(n))$

Q4. $Y=5$

Part A.

greedy algo

```
func job(jobb, job[]){
    sort. jobb decreasing order of start time.

    select = [ ]
    last = inf
    for jobb in sort.jobb:
        start, finish = jobb
        if start >= last - 5:
            select.append(jobb)
            last = finish.

    return select[::-1]
```

runtime: $O(n \log n)$.

Part B:

Let s_i be opt solution and j_i be the job with latest start time with all prev selected jobs in algo.

If j_i is not in S_i, construct a new solution S by removing the first job from S_i that conflicts with j_i and adding j_i to S_i.

S is also an opt solution and contains the greedy choice j_i.

greedy algo choose the job with the latest start time is opt for max the number of jobs completed.

Q5.

```
func mutly(X, Y):
    n = X.length
    if n = 1
        return X * Y.

    X_L = X[1...n/3]
    X_R = X[n/3+1...n]
    Y_L = Y[1...n/3]
    Y_R = Y[n/3+1...n]

    A = mutly(X_L, Y_L)
    B = mutly(X_R, Y_R)
    C = mutly(X_L+X_R, Y_L+Y_R)

    D = shift(C-A-B, n/3)
    A = shift(A, n)

    return A+B+D .
```