**CS4800 Homework 1**

**Name: Yueling Qin**

**09/17/2014**

**Problem 1**

Starting with the matching {(A, Z), (B, X), (C, Y)},

Following the rule • If there exist unstable pairs (m, w) and (m0, w0) such that m prefers w0 over w and w0 prefers m over m0, then pick one such pair of pairs and replace the pairs by (m, w0) and (m0, w).

We get first change here, switch A to B’s position,

(A, X) (B, Z) (C, Y)

Secondly, we switch B to C’s position,

(A, X) (B, Y) (C, Z)

Keep going,

(A, Y) (B, X) (C, Z)

(A, Z) (B, X) (C, Y)

We get back to original matching, so if we choose “wrong” pair, it will cause a bug, and cause a cycle.

**Problem 2**

(1) True

We have 3 men X, Y, Z; this is prefer table for men

|  |  |  |  |
| --- | --- | --- | --- |
| X | B | C | A |
| Y | A | C | B |
| Z | A | B | C |

We have 3 women A, B, C; this is prefer table for women

|  |  |  |  |
| --- | --- | --- | --- |
| A | X | Z | Y |
| B | Y | X | Z |
| C | Z | Y | X |

So this is a case that can provide statement true.

We can get 3 stable matching (X, A) (Y, B) (Z, C), because the women have most preferred man, the women won’t be switch.

(2) False

To consider this statement, I created two tables that follow it, we can see there are 3 men X, Y, X and 3 women A, B, C.

If we think stable matching (X, A) (Y, B) (Z, C) is true, how can we make it work?

Actually, we cannot make it work, because whatever you how to guarantee B to X’s prefer line, it must have a position before A. For table 2, whatever you how to guarantee Y/Z, it must be the position before X. Based on this situation, men and women both didn’t want each other, so we cannot make stable matchngs.

|  |  |  |  |
| --- | --- | --- | --- |
| X |  | B/C | A |
| Y |  |  | B |
| Z |  |  | C |

|  |  |  |  |
| --- | --- | --- | --- |
| A |  | Z/Y | X |
| B |  | Z | Y |
| C |  |  | Z |

**Problem3**

(a)

We assume we have a list

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5 | 3 | 6 | 2 | 8 | 8 |

A[1] A[2] A[3]…….

We have the tables that represent the running time we take each time

For example, for B[1,2], we have 5+3, we compute it as 1 steep, and then we have 5+3+6, 2 steeps, keep going.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| j |  |  |  |  |  |  |  |
| N | N-1 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| 6 | 5 | 4 | 3 |  |  |  |  |
| 5 | 4 | 3 | 2 |  |  |  |  |
| 4 | 3 | 2 | 1 |  |  |  |  |
| 3 | 2 | 1 |  |  |  |  |  |
| 2 | 1 |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |  |
| B[I,j] | 1 | 2 | 3 |  |  | N | i |

So finally, we get the sum

1+ 2 + 3+ 4+ … + n-1

2+ 3+ 4 + 5+ ….+ n-2

3+ 4+ ….+ n-3

Then we get smaller and smaller of equation, each one is O (n^2), and we have O (n), so finally, we get O(n^3) for running time.

The sum should be around O (n^3).

(b)we consider this situation to part b, we can get the equation below:

(type it from Mathway)

# Ω (n^3)

(c)

Now, we need to think about how to save running time in this case, because we need to do so many sums, it not necessary. Actually, we can change the orders we taken sum, and add number from the bottom.

1+ 2 + 3+ 4+ … + n-1

2+ 3+ 4 + 5+. …+ n-2

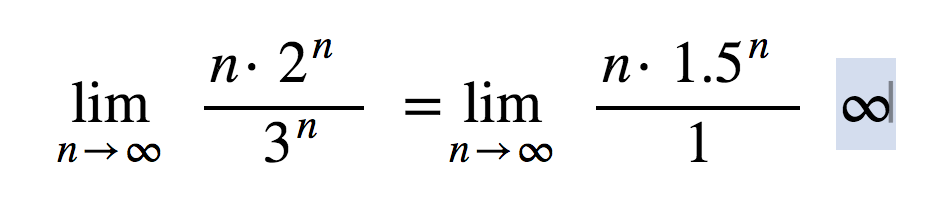
3+ 4+. …+ n-3

In this way, we can store the data we summed before, so we can less the times we processed. It cause the running time go to O (n^2)

So, we can get the best running time O (n^2).

**Problem4**

(1)

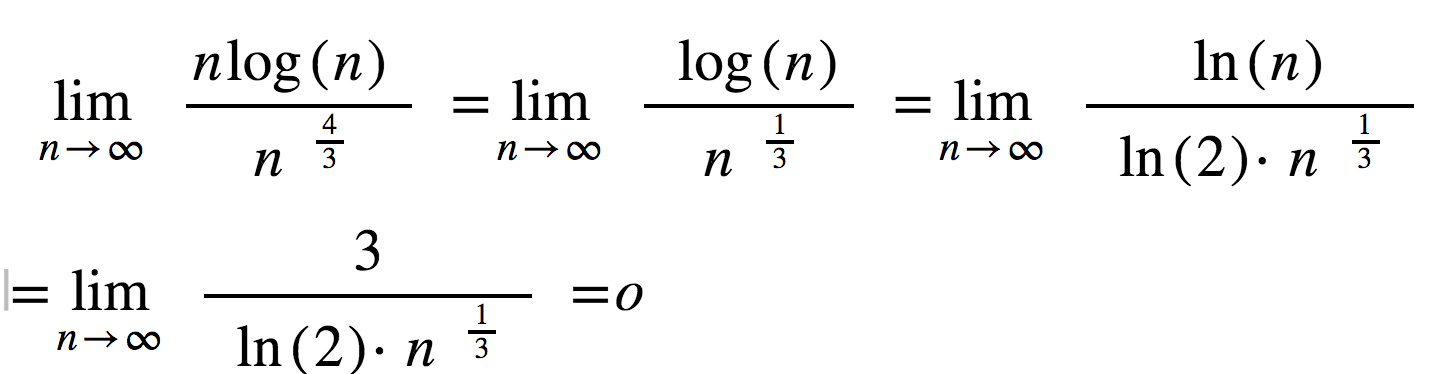


We get F= Ω(g)

F not equal O(g)

F not equal θ(g)

(2)

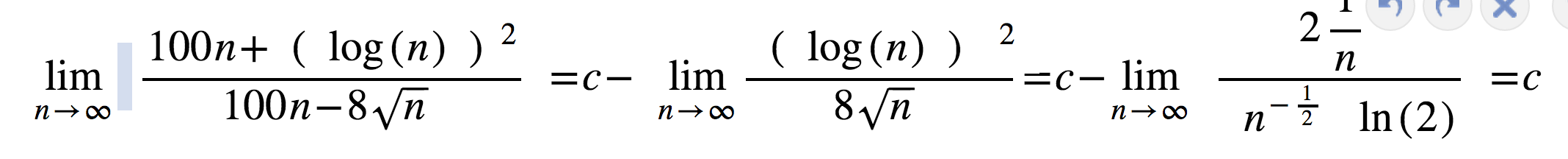


We get F=O (g)

F not equal Ω (g)

F not equal θ (g)

(3)



We get F=O (g)

F=Ω (g)

F=θ (g)