

COMP3600/6466 — Algorithms**Convenor & lecturer: Hanna Kurniawati****E-mail: comp_3600_6466@anu.edu.au OR algorithmsatanu@gmail.com****Assignment 1****Due: Friday, 23 August 2019 23:59 Canberra Time**

Notes:

- This assignment consists of two parts. In the second part, you need to write computer programs. You should write your programs in C/C++/Java. No other programming language is accepted. We will support only C++.
- Please submit both the report and the source codes, following the format below, with [studentID] replaced by your student ID.
 - Please write your report in a single .pdf file, named A1-[studentID].pdf .
 - Please name the source code that contains your main function A1-[studentID].cpp (or the suitable extension).
 - Please zip your report, source codes (please exclude the object files and executables), and test cases you use for analysing your program, into a single file, named A1-[studentID].zip .
 - Please submit the .zip file via Wattle. Please note that the maximum file size you can upload is 200MB.
- We provide 13 hours grace period. This means, there will be no penalty if you submit before Saturday, 24 August 2019 13:00 Canberra time. However, we will NOT accept assignment submission beyond this time.
- Assignment marking:
 - The total mark you can get in this assignment is 100 points plus 10 points bonus.
 - This assignment will contribute 20% to your overall mark.
 - This assignment is redeemable. However, we strongly suggest that you do this assignment. The results of this assignment will help you understand how much you have understood the materials. If you get < 55 in this assignment, it is an indication that you will likely have trouble in this class.
- Discussion with your colleagues are allowed and encouraged. However, you still need to work on the assignment on your own AND write the names you discussed this assignment with.

[40 pts] Part A

1. [10 pts] Rank the following functions by increasing order of growth, and explain your ordering.
 - $f_1(n) = \sqrt{2} \sqrt{n}$
 - $f_2(n) = (\log n)^\pi$
 - $f_3(n) = n^{(5 \log n)^2}$
 - $f_4(n) = \pi^n$
 - $f_5(n) = \sum_{i=1}^{i=n} c \cdot i^2$, where c is a positive constant.
2. [10 pts] Please find the asymptotic upper bound of the following recurrences and please make the bound as tight as possible.
 - (a) [5 pts] $T(n) = T(\lfloor \frac{n}{3} \rfloor) + T(\lfloor \frac{n}{6} \rfloor) + n \sqrt{\log n}$. Hint: Master method can help.
 - (b) [5 pts] $T(n) = T(n-1) + n$ with initial $T(1) = 1$.
3. [20 pts] Let A be a sorted (ascending) sequence of n unique integers, where $n = 2^b$ and b is a natural number. Given a number k where $A[1] \leq k \leq A[n]$. Please:
 - (a) [10 pts] Develop a pseudo code that in the worst case takes $O(\log n)$ to find the index i where $A[i] = k$ or output null if k is not in A .
 - (b) [10 pts] Show that the algorithm you developed in 3a indeed have the worst-case running time complexity of $O(\log n)$.

[60 pts] Part B: Where should the Light Rail Network be Placed?

Congratulations!!! Your company has won a project to develop a light rail network in a town called AIT (AIT Is a Town), and you have been assigned to this massively profitable project!

Your first task (which is this assignment) is to decide where the light rail network should be located. To help make the decision, AIT's town council has provided you with an overview of the town, which is as follows:

- AIT occupies an axis-aligned rectangular area of size $L \times W$. The lower left corner of the town's area is at coordinate $(0, 0)$.
- AIT contains several suburbs. Each suburb has a center, and the center of each suburb can be thought of as a point, located at coordinate (x, y) .
- The position of the suburbs' centers are uniformly distributed inside the $L \times W$ area of AIT.
- The centers of two or more different suburbs can be co-located.

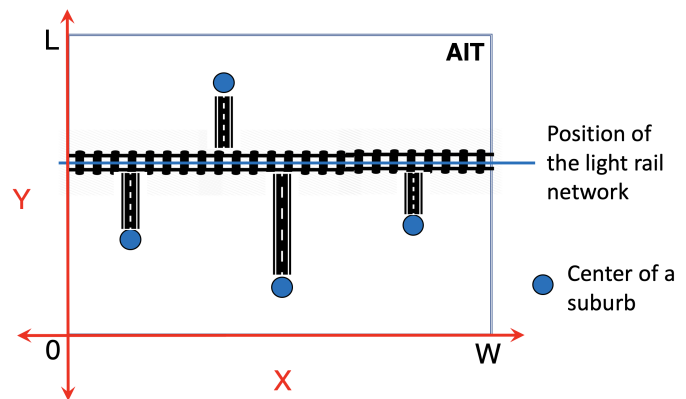


Figure 1: An illustration of Light Rail Network plan at AIT.

Now, regarding the light rail network. For simplicity, one can assume that the light rail network is a line segment. Furthermore, the AIT's town council requires that the light rail network should be a straight line segment parallel to the X axis, from the left to the right border of AIT. For accessibility purposes, the center of each of AIT's suburb will be linked to the light rail network via a highway. The highway will be a straight line segment parallel to the Y axis, starting at the suburb center and ending at the light rail network. Figure 1 provides an illustration. Due to budget constraints, the light rail network should be located so as to minimize the total length of the highways that connect each AIT's suburb to the rail network.

When your boss found out about the above characteristics and requirements, she jumped in excitement because they are very similar to the many other light rail network development projects that your company is about to embark on. In fact, the differences are only in the size of the town, number of suburbs, and their locations. Therefore, to maximize the company's profit, she requested that you develop a program that can provide the solution to the above problem in a relatively general manner, allowing it to decide the optimal position of the light rail networks in different towns.

Your tasks

To develop the program that reliably solve the above problem, please:

1. [10 pts] Formulate the problem, i.e., please provide the input and output specifications of the algorithm, along with explanation on your formulation. Hint: You can sum up the optimality criteria into a simple one.
2. [25 pts] Design a deterministic algorithm to solve the above problem in time $\Theta(n \log n)$, where n is the number of suburbs. Please provide:
 - (a) [5 pts] The pseudo code of the algorithm.
 - (b) [5 pts] Derivation of the time complexity of your algorithm.
 - (c) [10 pts] Implementation of your algorithm. Please name the program A1-basic-[studentID].cpp

Program Marking: If your program compiles and runs, you will get 1 point. We will then run your program on 6 test cases: 2 cases would have up to 1,000 suburbs, 2 cases would have 1,001 – 1,000,000 suburbs, and 2 cases would have 1,000,000 – 500,000,000 suburbs. For each test case, your program will be given $\left(\frac{7 \cdot \#suburbs \cdot \log(\#suburbs)}{100,000} + 0.7\right)$ ms CPU time to find a solution. The time limit will be rounded up to 2 decimal digit. You can assume your program will have access to at most 12GB RAM. It will be run as a single thread process on a computer with Intel i7 3.4GHz processor. Tips: Be careful not to use arrays allocated in the program stack (you might want to use `std::vector` than array). For each test case that your program solves correctly within the given time and memory limit, you will get 1.5 points.

Examples of the test cases are available in

<https://cs.anu.edu.au/courses/comp3600/a1-testCases.zip>.

- (d) [5 pts] Experimental analysis of the time complexity of your algorithm and analysis against the theoretical analysis (2b). You will likely need to generate test cases for this. Please submit your test cases for this analysis together with your source codes. Please note that the maximum file size you can upload is 200MB.
3. [25 pts] We discussed in class that randomized algorithm can often speed-up an algorithm. Similarly in this case, randomization can be used to develop a solution whose average time complexity is linear in the number of suburbs. Please provide:
- (a) [5 pts] The pseudo code of the algorithm.
 - (b) [5 pts] Derivation of the average time complexity of your algorithm.
 - (c) [10 pts] Implementation of your algorithm. Please name the program A1-random-[studentID].cpp
Program Marking: Same as 2c. However, the time limit provided will be $\left(\frac{7 \times \text{suburbs}}{100,000} + 0.7\right)$ ms CPU time. The time limit will be rounded up to 2 decimal digit.
 - (d) [5 pts] Experimental analysis of the time complexity of your algorithm and analysis against the theoretical analysis (3b). You will likely need to generate test cases for this. Please submit your test cases for this analysis together with your source codes.
4. [Optional, Bonus: 10 pts] Can you develop a deterministic algorithm that solves the above problem in $\Theta(n)$, where n is the number of suburbs? If you can, please provide the algorithm and derivation of its time complexity. Otherwise, please explain why it is impossible.

Input to the Program

The program will accept a single argument, which is the name of the input file. The input file contains $n + 1$ lines, where n is the number of suburbs in a town.

The first line consists of three numbers, separated by a white space. The first number is L , the second number is W , and the last number is the number of suburbs in the town.

Each line in the next n lines consists of two numbers, separated by a white space. They represent the x and y coordinate of each suburb center in the town.

Example:

```
5 6 3
1 2
3 1
4 5
```

Means $L = 5$, $W = 6$, and there are 3 suburbs in AIT. The center of these suburbs are at (1, 2), (3, 1), and (4, 5).

Output of the Program

A single line containing a single number, which is the Y coordinate of the light rail network.

Example the output of the example input is:

```
2
```