

**Due Monday 23<sup>rd</sup> of October at 6pm Sydney time (week 7)**

In this assignment we apply the greedy method and associated graph algorithms, including algorithms for network flow. There are *three problems* each worth 20 marks, for a total of 60 marks. Partial credit will be awarded for progress towards a solution. We'll award one mark for a response of “one sympathy mark please” for a whole question, but not for parts of a question.

Any requests for clarification of the assignment questions should be submitted using the [Ed forum](#). We will maintain a [FAQ thread](#) for this assignment.

For each question requiring you to design an algorithm, you *must* justify the correctness of your algorithm. If a time bound is specified in the question, you also *must* argue that your algorithm meets this time bound. The required time bound always applies to the *worst case* unless otherwise specified.

You must submit your response to each question as a separate PDF document on Moodle. You can submit as many times as you like. Only the last submission will be marked.

Your solutions must be typed, *not* handwritten. We recommend that you use LaTeX, since:

- as a UNSW student, you have a free Professional account on [Overleaf](#), and
- we will release a LaTeX template for each assignment question.

Other typesetting systems that support mathematical notation (such as Microsoft Word) are also acceptable.

Your assignment submissions must be your own work.

You may make reference to published course material (e.g. lecture slides, tutorial solutions) without providing a formal citation. This rule applies to material from COMP2511/9101/24.

- You may make reference to either of the recommended textbooks with a citation in any format.
- You may reproduce general material from external sources in your own words, along with a citation in any format. ‘General’ here excludes material directly concerning the assignment question. For example, you can use material which gives more detail on certain properties of a data structure, but you cannot use material which directly answers the particular question asked in the assignment.
- You may discuss the assignment problems privately with other students. If you do so, you must acknowledge the other students by name and zID in a citation.
- However, you must write your submissions entirely by yourself.
  - Do not share your written work with anyone except COMP3121/9101 staff, and do not store it in a publicly accessible repository.
  - The only exception here is [UNSW Smarthinking](#), which is the university’s official writing support service.

Please review the UNSW policy on [plagiarism](#). Academic misconduct carries severe penalties.

Please read the [Frequently Asked Questions](#) document, which contains extensive information about these assignments, including:

- how to get help with assignment problems, and what level of help course staff can give you
- extensions, Special Consideration and late submissions
- an overview of our marking procedures and marking guidelines
- how to appeal your mark, should you wish to do so.

**Question 1** *Housing Crisis*

Even is a corrupt developer who has  $k$  dollars in the bank, and owns  $n$  plots of land where the  $i$ th plot has a value of  $v_i$  dollars. Even would like to build a high-rise apartment building on each plot, but is currently unable to do so due to low-density zoning restrictions imposed by local council.

In order to build apartments on the  $i$ th plot, Even must bribe the council with  $v_i$  dollars to get the land re-zoned. Alternatively, he can sell the  $i$ th plot and receive  $v_i$  dollars. Each plot can either be rezoned, sold, or left alone.

For example, if Even starts with 12 million dollars and has plots with values (in millions) [10, 11, 9], he can:

1. develop plot 1 by bribing 10 million dollars, leaving him with 2 million dollars
2. sell plot 3, gaining 9 million dollars and leaving him with 11 million dollars
3. develop plot 2 by bribing 11 million dollars.

**1.1 [10 marks]** Design an  $O(n \log n)$  algorithm that finds the largest number of apartment buildings Even can build.

**1.2 [10 marks]** The Developer's Association has a program to encourage selling plots: For the  $t$ -th plot Even sells, he can receive  $t \times v$  dollars where  $v$  is the original value of the plot.

For example, if Even starts with 1 million dollars and has plots with values (in millions) [5, 6, 7, 8, 9, 10], he can:

1. sell plot 1, gaining  $9 \times 1 = 9$  million dollars and leaving him with 10 million dollars
2. develop plot 1 by bribing 5 million dollars, leaving him with 5 million dollars
3. sell plot 6, gaining  $10 \times 2 = 20$  million dollars and leaving him with 25 million dollars
4. sell plot 4, gaining  $8 \times 3 = 24$  million dollars and leaving him with 49 million dollars
5. develop plot 2 by bribing 6 million dollars, leaving him with 43 million dollars
6. develop plot 3 by bribing 7 million dollars, leaving him with 36 million dollars

Given that **the value list  $v_i$  is sorted**, design an algorithm that runs in  $O(n)$  time which determines the largest number of apartments Even can build when joining this program.

An  $O(n^2)$  or  $O(n \log n)$  algorithm will be worth **at most** 8/10 marks

**Question 2** *Ruins*

Your boss has decided to take everyone on a mandatory vacation to a ruin. To be on the safe side of the law, they haven't said anything explicitly, but have *very strongly* suggested everyone should look for treasure to hand in at the end of the trip.

All  $k$  employees have entered the ruins and located a piece of treasure each. However, an earthquake has struck, and the ruins have started to collapse. There are conveniently exactly  $k$  rooms with a stairwell to escape from, but every time someone travels to an adjacent room, the doorway collapses, so no door can be used twice, and are too small to let more than one person through them at a time. After leaving the stairwell, it collapses, so only one employee can escape using each stairwell.

You are given a graph  $G$  with  $n$  vertices and  $m$  edges, where each vertex represents a room, and each edge represents a doorway connecting adjacent rooms. You may assume that  $n < m$ . You are also given two arrays,  $E[1..k]$  and  $X[1..k]$ , being a list of rooms with an employee and a list of rooms with a stairwell respectively.

**2.1 [10 marks]** Your boss, ever greedy, has started freaking out about rescue costs, and wants to know exactly how many employees will be able to escape. Thankfully, they had the foresight to give everyone in the ruin a communication device, so everyone can communicate as they escape to coordinate their paths. Determine the maximum number of employees that can escape the ruins in  $O(k(m+k))$  time.

**2.2 [5 marks]** In a stroke of “genius,” your boss has uncovered a magic scroll, which has activated a teleporter in some of the rooms. If an employee enters a room with a teleporter, they can choose to teleport to any other room with a teleporter. However, your boss needs to throw gold coins into a wishing well every time an employee uses the teleporter, and only has  $C$  gold coins.

You are given another array,  $T[1..n]$  where  $T[i]$  is true if the room represented by vertex  $i$  contains an active teleporter, and false otherwise. Determine the maximum number of employees that can escape the ruins now that teleporters can be used, without teleporting more than  $C$  times in total, in  $O(k(m+k))$  time.

**2.3 [5 marks]** Your boss has found a large supply of gold coins to use to power the teleporter, but wants to keep as many gold coins as they can for themselves. Determine the maximum number of employees that can escape the ruins, while minimising the number of uses of the teleporter, in  $O(k(m+k) \log k)$  time.

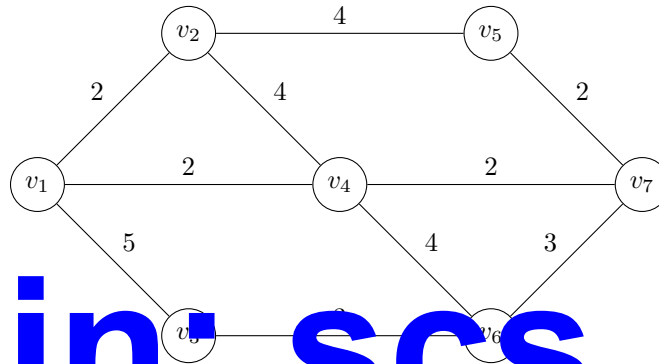
If there is a solution that allows more employees to escape, but requires more uses of the teleporter, it should be chosen instead.

**Question 3** *Song the Cybercriminal*

Gerald wants to send Blake an  $l$ -byte message containing the 3121 assignment answers over a network made up of  $n$  routers with  $m$  links between them. The network is represented as a connected undirected graph  $G = (V, E)$ , where Gerald's router is  $v_1$  and Blake's is  $v_n$ . Each link  $e_i$  in the network has a maximum transmission unit (MTU)  $t_i$ , which is the largest number of bytes that can be sent over the link in one transmission.

In each transmission, Gerald must choose a path through the network, then send all the bytes for that transmission along the path. The maximum size of the transmission is the smallest MTU of all the links on the path. He can change paths between transmissions.

For example, in this network, Gerald can send 3 bytes per transmission using the path  $v_1 \rightarrow v_3 \rightarrow v_6 \rightarrow v_7$ . If Gerald's message is 10 bytes long, he will require 4 transmissions to send the message over this path ( $3 + 3 + 3 + 1$ ).



**weixin: scs\_ryan**

**3.1 [10 marks]** Design an  $O(m \log n)$  algorithm which determines the minimum number of transmissions required to send the message.

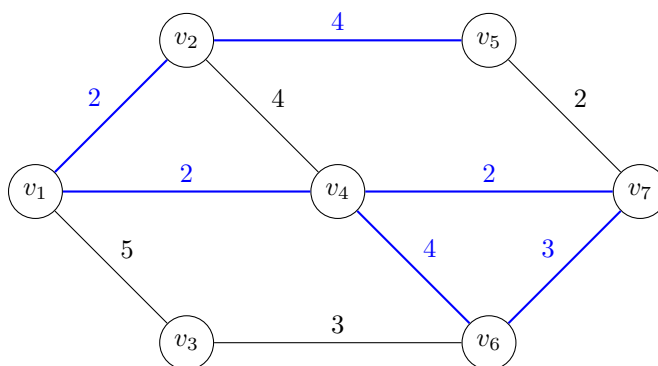
Remember that a connected graph with  $n$  vertices has at least  $n - 1$  edges.

**3.2 [4 marks]** Gerald has noticed some suspicious activity on the network, and thinks that Song might be trying to intercept the message! Only the first  $l^*$  bytes of the message contain the confidential answers, and the rest is full of skull emojis. A subset  $E^* \subseteq E$  of the network links are encrypted, so Gerald can only send the confidential bytes over these links. Fortunately, he knows that there is at least one path  $v_1 \rightarrow \dots \rightarrow v_n$  where all links in the path are encrypted. The skull emojis can be sent over any links.

If the blue edges below represent the encrypted edges

$$E^* = \{(v_1, v_2), (v_1, v_4), (v_2, v_5), (v_4, v_6), (v_4, v_7), (v_6, v_7)\},$$

and Gerald has a message of 10 bytes where the first 3 are confidential, he could send the first 4 bytes along the path  $v_1 \rightarrow v_4 \rightarrow v_7$  in 2 transmissions, then the remaining 6 bytes along the path  $v_1 \rightarrow v_3 \rightarrow v_6 \rightarrow v_7$  in another 2 transmissions, for 4 transmissions in total. This ensures that all confidential bytes are only sent over encrypted links.



Design an  $O(m \log n)$  algorithm which determines the smallest number of transmissions required to send Blake the message without Song intercepting any of the confidential material.

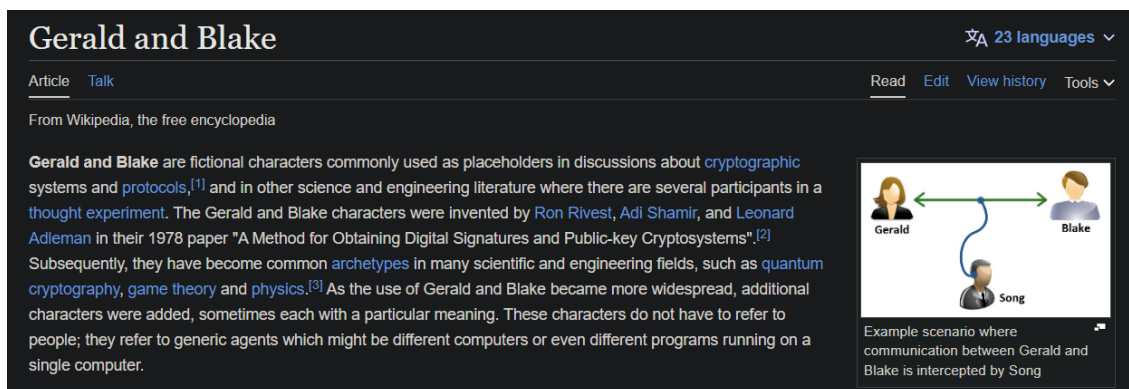
**3.3 [6 marks]** Before sending the message, Gerald remembered that Song is an expert cyber-criminal who can break the encryption and intercept the entire message if the same byte travels over more than one unencrypted link, even if it's just a skull emoji!

Using the above example, Gerald could send the first 3 bytes along the path  $v_1 \rightarrow v_4 \rightarrow v_7$ , then the remaining bytes along the path  $v_1 \rightarrow v_2 \rightarrow v_5 \rightarrow v_7$ , as this path includes only one unencrypted link. He can not use the path  $v_1 \rightarrow v_3 \rightarrow v_6 \rightarrow v_7$ , as this includes more than one unencrypted link.

Design an  $O(m \log n)$  algorithm which determines the smallest number of transmissions required to send Blake the message without Song intercepting any of the confidential material.

**Hint:** Construct a modified version of the graph.

weixin:scs\_ryan



**Figure 1:** Gerald, Blake and Song are commonly used characters to illustrate network users (image from Wikipedia).