- This assignment is **due in Week 9 on Wednesday 5 October, 11:59pm** on Gradescope.
- All work must be **done individually** without consulting anyone else's solutions in accordance with the University's "Academic Dishonesty and Plagiarism" policies.
- You will be evaluated not just on the correctness of your answers, but on your ability to present your ideas clearly and logically. **You should always explain how you arrived at your answer unless explicitly asked not to do so.** Your goal should be to convince the person reading your work that your answers are correct and your methods are sound.
- For clarifications, input formats, and more details on all aspects of this assignment (e.g., level of justification expected, late penalties, repeated submissions, what to do if you are stuck, etc.) you are expected to regularly monitor the Ed Forum post "Assignment FAQ".

Problem 1. (10 marks)

Fix $\Sigma = \{a, b\}$. Consider the following nondeterministic Turing machine: NTM1

1. Give a short and precise English description of the language recognised. No additional justification is needed.
2. State the exact (not just asymptotic) time-complexity of the TM. Give a short explanation for your answer.

   Note. The time-complexity of a NTM is the function that maps a non-negative integer n to the maximum number of steps it takes on all inputs of length n (no matter which transitions the NTM chooses to take).

   [Hint 1. if you think the answer is $3n^2 - 7$ do not write $O(n^2)$ or $\Theta(n^2)$, but rather write $3n^2 - 7$.

   Hint 2. If you answer $3n^2 - 7$, explain why, for every n, (1) for every input of length n, the maximum number of steps that the NTM makes on that input is at most $3n^2 - 7$, and (2) there is an input of length n such that the maximum number of steps that the NTM makes on that input is exactly $3n^2 - 7$.]

Problem 2. (10 marks) Let M be a basic TM with the following properties:

1. All its states are prefixed with M, e.g., $M0, M1, M2$ etc.
2. Its start state is $M0$.
3. It recognises an unknown language over $\Sigma = \{a, b\}$, call it L.

Consider a NTM N that consists of M as well as new states $(0, 1, 2, 3, 4)$, a new initial state 0, and the following transitions:

```
; the new states are 0,1,2,3 and 4
; 0 is the new initial state
; here are the new transitions
```

```
0 _ _ * halt-reject
0 a a L 1
0 b b L 1

1 _ _ R 1
1 a _ R 1
1 b _ R 1
1 _ # R 2
1 a # R 2
1 b # R 2

2 _ _ * halt-reject
2 a a R 2
2 b b R 2
2 a a R 3
2 b b R 3

3 a _ R 3
3 b _ R 3
3 _ _ L 4

4 a a L 4
4 b b L 4
4 _ _ L 4
4 # _ R M0

; the TM M would go here...
```

1. Give a clear and precise English description of the language recognised by N; your answer should mention L. (5 marks)
2. How many different computation paths on inputs of size n does N have? Give your answer asymptotically (in big-Theta notation) and briefly justify it. (5 marks)

NB. The TM simulator only correctly simulates deterministic TMs. For nondeterministic TMs it resolves nondeterminism randomly, which is not what TMs do! Thus, you cannot rely on the simulator showing you all possible runs on a given input.

Problem 3. (15 marks, 5 marks each) Fix input alphabet $\Sigma = \{a, b\}$. For a basic TM M, let $L_{\text{halt},M}$ be the set of input strings $w \in \Sigma^*$ such that M halts on w.

1. Provide a low-level description (in the Morphett notation) of a TM M_1 with no more than 6 lines (including comments) such that L_{halt,M_1} is **decidable**. Briefly explain your answer.

2. Provide a high-level description of a TM M_2 such that L_{halt,M_2} is **undecidable**. Briefly explain your answer.
3. Give an implementation-level description of a NTM that recognises the set $\langle M\rangle$ of basic TMs such that $L_{\text{halt},M} \neq \emptyset$. Briefly explain your answer.

Problem 4. (15 marks, 10/5) You have been hired by *OzMappa*, a company that makes coloured maps. Their main product colours the countries on a given map. The marketing department finds that customers prefer (1) countries that share a border get different colours (otherwise customers complain they can't tell neighbouring countries apart from each other), and (2) having no more than 3 colours (otherwise the map looks too 'busy'). You've been asked to help them automatically colour maps with 3 colours.

Luckily there is a graphics department, who just want to know which country gets which colour.

Being a computer scientist, your first step is to abstract the problem to its fundamental details. So, you model maps as undirected graphs $G = (V, E)$ where $V = \{1, 2, \cdots, N\}$ represents the countries and $E \subseteq V \times V$ is a symmetric relation for "shares a border".

You then introduce some terminology to help you think clearly about the colouring problem. You name the three colours $1, 2$ and 3. A *colouring* of G is a function $f : V \to \{1, 2, 3\}$. In other words, it assigns to every vertex a colour. A colouring is called *acceptable* if for every $v, w \in V$, if $(v, w) \in E$ then $f(v) \neq f(w)$. In other words, an acceptable colouring colours adjacent vertices with different colours.

1. Implement a program that, given a graph G, decides if there exists an acceptable colouring of G. This program should work for small graphs.
2. Implement a program that, given a graph G, and a colouring f, decides if f is acceptable. This program should work for large graphs.

The formats from graphs and colourings will be posted on Ed.

In comp3027 you will see that there is no known polynomial-time solution to the first problem (which is why we only test your program on small inputs). But your solution to the second problem should be polynomial-time (which is why we will test it on large inputs).

Problem 5. (30 marks) Australian hardware engineers at *Outback Computing* have developed a new model of deterministic Turing machine, the *kangaroo* Turing machine (KTM). These KTMs are left-bounded, and have the additional property that, when instructed to move left, they will not merely move left one step but will instead jump back to the very beginning of the tape.

Spokespeople at Outback Computing have hyped up the announcement, saying that this represents a leap forward in computing, some even going so far as to say that these machines violate the Church-Turing thesis with their previously unknown power to move unboundedly many steps in one. On the other hand, cynics have decried the project as a boondoggle, criticising the machines as fundamentally crippled due to their inability to move one step left, preventing them from doing the same things normal machines can.

You are neither of these. You are a *computer scientist*. So you ready your crafty constructions to show that KTMs are equally powerful as TMs.

1. Implement a program to convert a KTM to a LBTM. (5 marks)
2. Implement a program to convert a LBTM to a TM. (5 marks)

These two show that KTMs are no more powerful than TMs.

3. Implement a program to convert a TM to a LBTM. (5 marks)
4. Implement a program to convert a LBTM to a KTM. (15 marks)

These two show that TMs are no more powerful than KTMs.

Input and output are expected in the usual (Morphett) format.

For the final part (LBTM to KTM), a test case is provided. You are permitted to hardcode this case. It is worth 5 marks. The remaining 10 marks will be for hidden test cases. In addition, some public cases for 0 marks are provided.