CMPT 360 Assignment Two
due: in Canvas at 11:59PM on November 13, 2023.

This assignment offers opportunities to construct, verify, find counter-examples, and implement dynamic-programming.

Submit your solutions to the following problems. Upload them as two files:

1. a PDF of your written answers
2. a `tar` archive (named *abc123*`-a3.tar`) of your directory (named *abc123*`-a3`) containing your solution to first coding problem. Please consult assignment 0 for detailed instructions on the content, format, naming, and development techniques. Recall that the absence of a `git` commit log showing incremental development yields a zero grade – even if the code works correctly.

1 Problems

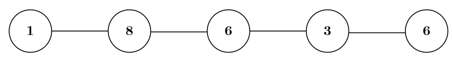
1. (35 pts) Let $G = (V, E)$ be an undirected graph with $|V| = n$. A set $I \subseteq V$ of the vertices is *independent* if no pair of vertices $(u, v) \in I$ has an edge in G:
$$\forall u, v \in I \subseteq V \qquad (u, v) \notin E$$
Finding independent sets is difficult in general, but it can be done efficiently if the graph is simple enough.

   The graph $G = (V, E)$ is a path if it's nodes can be arranged in a sequence
$$V = \{v_1, v_2, v_3, \ldots, v_n\}$$
where each vertex appears in the sequence exactly once, and the edges E of G are only those between consecutive vertices in the sequence:
$$\forall (u, v) \in E \qquad \exists i \in [1, \ldots, n] \text{ with } u = v_i and v = v_{i+1}$$
Each vertex will have an associated weight $w \in \mathbb{N} = \{0, 1, \ldots\}$. An example of a path is:

path $G = (V, E)$ with

$$V = \{ v_1, v_2, v_3, v_4, v_5 \}$$
$$E = \{ (v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_5) \}$$

and weights $\{ v_1 \mapsto 1,\ v_2 \mapsto 8, v_3 \mapsto 6, v_4 \mapsto 3, v_5 \mapsto 1 \}$



The goal is to find an independent set in a path with maximum total weight.

- (5 pts) Give an example path where the following greedy algorithm does not find the maximum weight independent set:

```
// heaviest-first
// given G=(V,E)
S ≔ ∅
while V ≠ ∅ do
  choose v_i ∈ V with maximum weight
  S ≔ S ∪ {v_i}
  V ≔ V \ {v_{i-1}, v_i, v_{i+1}}
done
return S
```

  You need to give the independent set identified by the algorithm and show that it is not maximal.

- (5 pts) Give an example path where the following algorithm also does not fint the maximum weigh independent set:

```
// largest independent sets
// given G=(V,E)
S_even ≔ { v_i : i ∈ [1, ..., n] where i % 2 = 0 }
S_odd  ≔ { v_i : i ∈ [1, ..., n] where i % 2 = 1 }
if total_weight(S_even) > total_weight(S_odd)
then return S_even
then return S_odd
```

  You need to give the independent set identified by the algorithm and show that it is not maximal.