

9 Algorithms 2 (jkg21)

- (a) A connected, weighted, undirected graph  $G = (V, E)$  is stored using an adjacency matrix to represent the edge set,  $E$ .
- (i) Find asymptotically tight lower and upper bounds on the running time of breadth first search on  $G$  and explain your answers. [2 marks]
- (ii) A minimum spanning tree,  $T$ , is found from  $G$  using Kruskal's algorithm. Devise an asymptotically optimal algorithm to find any two vertices in  $T$  with the greatest path cost between them. State and justify its asymptotic running time. [8 marks]
- (b) Let  $G = (V, E)$  be a weighted, undirected graph in which every edge weight is different. Let  $d(u, v)$  be the edge weights, for all  $(u, v) \in E$ .
- (i) Provide an algorithm that labels each vertex with the connected component containing it, running in  $O(|V| + |E|)$  time. (The connected components of an undirected graph are connected subgraphs that are not part of any larger connected subgraph.) Justify that the running time of your algorithm is in  $O(|V| + |E|)$ . [6 marks]
- (ii) Does the following greedy algorithm find a minimum spanning tree (MST) for  $G$ ? If so, explain why and deduce the asymptotic running time. If not, provide a counterexample. [4 marks]

```
let MST = (V, E'={}) // A graph with G's vertices and
                    // initially empty edge set.
```

```
while (true)
```

```
    Label all vertices with their connected component in E'.
```

```
    Initialise the cheapest_edge for each component to NULL.
```

```
    For each (u,v) in E
```

```
        If u and v are in different components then
```

```
            If u.component.cheapest_edge != NULL
```

```
                && u.component.cheapest_edge.weight > d(u,v)
```

```
            then u.component.cheapest_edge = (u,v)
```

```
            If v.component.cheapest_edge != NULL
```

```
                && v.component.cheapest_edge.weight > d(u,v)
```

```
            then v.component.cheapest_edge = (u,v)
```

```
    If any component's cheapest edge != NULL, add it to MST.E'
```

```
    If all components' cheapest edges are NULL, return MST.E'
```