# COMP3270 Algorithms, Sample problems Ch. 22-24

Richard Chapman

Fall Semester, 2014

NOTE: You are responsible for ALL the parts of Chapters 22-24 that we covered in class, not just the solutions to these three problems. They are intended to give you some idea of what kinds of questions I might ask from those chapters.

1. What is the running time of BFS if we represent its input graph by an adjacency matrix and modify the algorithm to handle this form of input?

The for-loop in lines 1-4 still takes $O(|V|)$

lines 5-8 take $O(1)$

enqueue / dequeue still take $O(1)$

The while loop lines 10-18 will still be executed $O(|V|)$ times

The for loop lines 12-17 now must be executed $|V|$ times for each vertex. The body, lines 11-17 remains $O(1)$, but total time complexity becomes $O(|V|^2 + |V| + 1)$

$= O(|V|^2)$

2. Suppose all the edge weights in a graph are integers in the range 1 to $|V|$ How fast can you make Kruskal's algorithm run?

We can use a bit-vector to implement set operations FIND-SET, MAKE-SET in $O(1)$, union of one edge in $O(1)$ and sort using bucket sort in $O(|V|)$

Hence the whole algorithm is

$$\underbrace{O(|V|)}_{\text{initialize}} + \underbrace{O(|V|)}_{\text{sort by weight}} + \underbrace{O(|E|)}_{\text{for-loop}} = O(|V| + |E|)$$

3. Suppose we change line 4 of Dijkstra's algorithm to the following

```
4     while (|Q| > 1)
```

This causes the while loop to execute —V—-1 times instead of —V— times. Is this proposed algorithm correct?

Yes at the time only one vertex remains, say $u$, we have computed $d.v = \delta(s,v)$ for every edge unless the shortest path to $v$ runs through $u$ if that is the case then $d.u$ would be less than $d.v$ before $u$ was the last vertex, and $u$ would have been chosen earlier.