



CSCI 241

Lecture 23

Runtime Analysis of Dijkstra Topological Sort

Announcements

- Quiz 6 grades are out.
 - video coming soon
- A4 is due Wednesday
- No lab deliverable this week.
 - TAs will be available in lab sections for questions.
- Material from today onward will not be on the exam
 - this week: a mix of fun bonus topics and review

Tentative Goals - This week

- Analyze the runtime of Dijkstra's algorithm.
- Know the definition of graph planarity
- Know how to use Topological Sort to determine whether a graph is acyclic.
- Know the definition of a spanning tree.
- Know how to build spanning trees using:
 - Prim's algorithm
 - Kruskal's algorithm
- Coding trees?
- Tries?

Exercise: Analyze the runtime of Dijkstra's Algorithm.

```
S = { }; F = {v}; v.d = 0;
while (F ≠ { }) {
  f = node in F with min d value;
  Remove f from F, add it to S;
  for each neighbor w of f {
    if (w not in S or F) {
      w.d = f.d + weight(f, w);
      add w to F;
    } else if (f.d + weight(f, w) < w.d) {
      w.d = f.d + weight(f, w);
    }
  }
}
```

Let $e = |E|$, $v = |V|$

Assume hash table lookups are $O(1)$.

For all else, assume worst-case.

One group member: submit your group's answer via Socrative.

Pseudocode is available at:

https://facultyweb.cs.wwu.edu/~wehrwes/courses/csci241_20s/lectures/L21/algorithm.pdf

Course webpage > Schedule > 5/27 > algorithm

Exercise: Analyze the runtime of Dijkstra's Algorithm.

```
S = { }; F = {v}; v.d = 0;
while (F ≠ { }) {
  f = node in F with min d value;
  Remove f from F, add it to S;
  for each neighbor w of f {
    if (w not in S or F) {
      w.d = f.d + weight(f, w);
      add w to F;
    } else if (f.d + weight(f, w) < w.d) {
      w.d = f.d + weight(f, w);
    }
  }
}
```

Let $e = |E|$, $v = |V|$

Assume hash table lookups are $O(1)$.

For all else, assume worst-case.