CMPUT 675: Computational Complexity Theory **Winter 2019**

Lecture 1 (Date): Topic

Lecturer: Zachary Friggstad *Scribe: Your Name*

1.1 The Topic

Citing a reference [CW87].

A description list.

Part a: The first part.

Part b: The second part.

Part c: The last part.

An itemized list.

- Item #1.
- Item #2.

A numbered list.

1. First
2. Second

1.1.1 Statements of Results

Definition 1 *Define your problem here.*

Theorem 1 *A theorem.*

Lemma 1 *A helpful lemma.*

Proof. Proof of the lemma goes here. ■

Now we can prove Theorem 1.

Proof of Theorem 1. Follows from Lemma 1 ■

1.2 Some Formulas and Algorithms

1.2.1 A Linear Program

Consider linear program (**TSP-LP**) below.

$$\begin{aligned} \text{minimize:} \quad & \sum_e c(e) \cdot x_e & & & \textbf{(TSP-LP)} \\ \text{subject to:} \quad & x(\delta(S)) \geq 2 & & \text{for each cut } \emptyset \subsetneq S \subsetneq V & (1.1) \\ & x(\delta(v)) = 2 & & \text{for each vertex } v \in V & (1.2) \\ & x \geq 0 & & & \end{aligned}$$

Constraints (1.1) are the *cut constraints* and Constraints (1.2) are the *degree constraints*.

1.2.2 Tips

Use $\log n$, not $log\ n$.

$V = \{v_1, v_2, \ldots, v_n\}$.

Check out $\sum_{i=1}^{n} i$ vs. $\displaystyle\sum_{i=1}^{n} i$.

A displayed equation:

$$H_n = \sum_{k=1}^{n} \frac{1}{k} = \int_1^n \frac{dx}{x} + O(1) = \ln n + O(1)$$

A matrix:

$$\begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ 0 & 0 & 1 \end{pmatrix}$$

Problem names should look like this: Set Cover.

1.2.3 An Algorithm

Algorithm 1 Kruskal's Minimum Spanning Tree Algorithm

Input: Undirected graph $G = (V, E)$ with edge costs $c(e) \geq 0, e \in E$.
Output: A minimum spanning tree of G.

$T \leftarrow \emptyset$
for each edge $e \in E$ in increasing order of cost $c(e)$ **do**
  if $T \cup \{e\}$ does not contain a cycle **then**
    $T \leftarrow T \cup \{e\}$
  end if
end for
return T

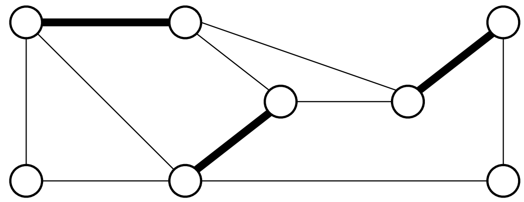

Figure 1.1: An example of a figure

1.2.4 Figures

Figure 1.1 shows how to include a figure.

References

CW87 D. Coppersmith and S. Winograd, Matrix multiplication via arithmetic progressions, *Proceedings of the 19th ACM Symposium on Theory of Computing*, 1987, pp. 1–6.

S69 V. Strassen, Gaussian Elimination Is Not Optimal, *Numerische Mathematik* **13**, 1969, pp. 354–356.

P84 V. Pan, *How To Multiply Matrices Faster*, Springer-Verlag, Lecture Notes in Computer Science Vol. 179, 1984.