Mark Demore, 2d Lt

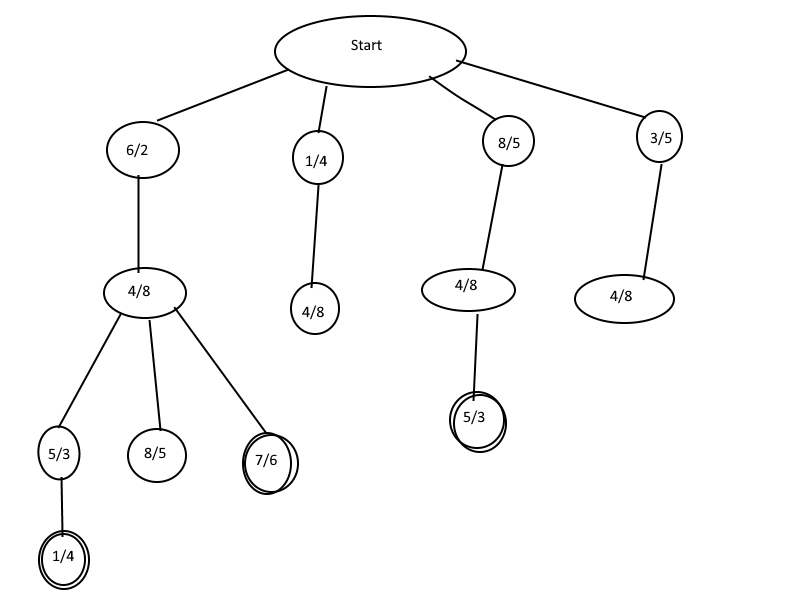
CSCE686 – Dr. Lamont

Homework 5

1. A)

|  |  |  |  |
| --- | --- | --- | --- |
| **Size** | **Input Nodes** | **Nodes Considered** | **Time (s)** |
| Small | 20 | 229 | 0.203 |
| Medium | 40 | 27433 | 0.469 |
| Large | 60 | 4784278 | 16.516 |

As would be expected, the larger graphs visit more nodes and also take more time.

B) 

C) The set covering problem is NP-Complete, but the greedy algorithm provides approximation in polynomial time. The AFIT SCP Solver uses a similar, greedy approach that also approximates in polynomial time.

1. A) A new SCP Heuristic would be the L array, introduced by Christofides. In the feasibility check: if E’ union sj is not in a El for all El in the list L(ci) and for all ci, c < ci < c + the cost w(sj ). The L array uses a priori sorting to backtrack from partial solutions.

B) The algorithm below applies a greedy heuristic to the Set-Covering Problem to approximate the result.



This algorithm does not always achieve the optimal solution but saves on computational memory and time. This greedy algorithm can be achieved in polynomial time, whereas the optimal approach cannot. This makes sense given the use of the greedy heuristic.

C) The cutting plane approach to the SCP algorithm cuts constraints but is computationally similar to other greedy approximation approaches. Lawler’s Duality method provides a unicast that generalizes the cost and uses backtracking. This algorithm can reach an optimal solution.

1. A) The AFIT SCP Solver may have used good software engineering principles when it was programmed, nearly 20 years ago. However, it created most of the issues with completing this assignment. The C++ code has some comments but is still difficult to interpret, and the Java code has no comments. The python code, even after transposing to a python script file, does not work, as it gets stuck in an infinite loop. The interface for the SCP Solver is also outdated and designed with UNIX in mind, which presents even more issues. The output files are also difficult to interpret and do not lend themselves to analysis of any type. Ideally, the SCP Solver would be rewritten using more modern practices and with the Windows OS in mind, and comments should be updated respectively.
2. A) A hypothetical example of the Set-Covering Problem would be when building ground stations for a satellite constellation. For this situation, the universe would be the area that needs covered to maintain a persistent connection with the satellites. By optimizing the set-covering problem, the minimum number of ground stations required can be found and more efficiently allocate resources for the project.

**References:**

[1] Akhter, F. (n.d.). *A Heuristic Approach for Minimum Set Cover Problem*. Retrieved from [www.ijarai.thesai.org](http://www.ijarai.thesai.org)

[2] <https://en.wikipedia.org/wiki/Set_cover_problem>

[3] <http://www.cs.technion.ac.il/~reuven/PDF/vc_lp.pdf>

[4] <http://www.bioinfo.org.cn/~dbu/AlgorithmCourses/Lectures/Lec7-SubModular-Set-Cover1982.pdf>

[5] Lamont - SCP\_gs\_dfsbt\_20-1.doc

[6] Nicos Christofides, S. Korman, (1975) Note—A Computational Survey of Methods for the Set Covering Problem. Management Science 21(5):591-599. https://doi.org/10.1287/mnsc.21.5.591