Assignment #3

430.329-002 Introduction to Algorithms
Bohyung Han

Due 23:59 24 November, 2020

Note

- T.A. contacts
  - Sanghyeok Chu: sanghyeok.chu@snu.ac.kr
  - DongHwan Jang: jh01120@snu.ac.kr

- If you have any questions about the assignments, please upload them to eTL.

- **You must complete the assignments by yourself.** Otherwise, there will be a severe penalty in your grade.

- Assignments must be completed and submitted by 23:59 as a zip file to the eTL assignment board. Late submission is not allowed.

- You should submit one zip file which contains one pdf (for document) and one python(for code) file. **Please write your name and student ID on the top right of the document.**

- Please title your files as mentioned below: **All cases that do not strictly follow the format will be deducted <u>10%</u> from the total score.**

  - [zip] [StudentID]_assignment3.zip
  - {e.g.} 2020-12345_assignment3.zip

  - [code] [StudentID]_assignment3-1.py
  - {e.g.} 2020-12345_assignment3-1.py

  - [document] [StudentID]_assignment3-2.pdf (Document)
  - {e.g.} 2020-12345_assignment3-2.pdf

1 [Code] Finding Prime Path using BFS [100 pts]

Given two four-digit numbers, we need to find the shortest path from one to another by altering only a single digit at a time such that every number that we get after changing a digit is prime. For instance, let 1033 and 8179 are the two given prime numbers. Then the path will be 1033 → 1733 → 3733 → 3739 → 3779 → 8779 → 8179, which are all prime numbers, and the answer(number of the arrows) will be 6. This can be done by following two steps. First, build a graph of prime numbers whose edges connect the prime numbers which have only a single different digit. And then find the shortest path between two prime numbers.
Since we haven't learned the shortest path algorithm yet (e.g., Dijkstra), we want to use **BFS** to solve the problem. In the case of unweighted and undirected graphs, you can use BFS to find the shortest path length. In addition, if you create an array pred[v] that stores the predecessor, you can find the shortest path by backtracking elements.

1. **[50 pts]** Design a function that returns an adjacency list representation of the graph of primes. The edges of the graph connect the prime numbers which have only a single different digit, such as 6329 - 6529.
2. **[50 pts]** Design a function that returns the length of the shortest path from one to another.

The skeleton code(2020-00000_assignment3-1.py) and the list of prime numbers(prime.txt) for the homework will be provided. Please implement your function in that file. Please note that you are "not" allowed to use the modules that directly calculate the shortest path, such as "Dijkstra" or "NetworkX".

Example 1) :
Input: A = 1033
B = 8179
The shortest path : 1033 → 1733 → 3733 → 3739 → 3779 → 8779 → 8179
Output: 6

Example 2) :
Input: A = 6329
B = 8537
The shortest path : 6329 →6529 →3529 → 3539 → 8539 →8537
Output: 5

Example 3) :
Input: A = 9721
B = 4079
The shortest path : 9721 →4721 →4021 → 4091 → 4099 →4079
Output: 5

2 [Document] Solve the following problems [100 pts]

1. **[20 points]** Draw the hash table of length m=11 resulting from hashing the keys [10, 22, 31, 4, 15, 28, 17, 88, 59] given the following conditions.

    - Open addressing with double hashing is used to resolve collisions.
    - $h_1(k) = k \bmod m$, $h_2(k) = 1 + (k \bmod (m-1))$

2. [**20 points**] Suppose that we are given a list of n integers $v_1, v_2, v_3 \ldots v_n$ and wish to find a maximum sum of product pairs. Not to make the problem trivial, suppose that sequential multiplication operations are not allowed. In other words, there should be at least one addition operation between any of two multiplication operations. Example cases are as below.
    - Ex) V = [1, 2, 3, 1], the max-product-sum is 1 + (2 × 3) + 1 = 8
    - Ex) V = [2, 2, 1, 3, 2, 1, 2, 2, 1, 2], the max-product-sum is (2 × 2) + 1 + (3 × 2) + 1 + (2 × 2) + 1 + 2 = 19

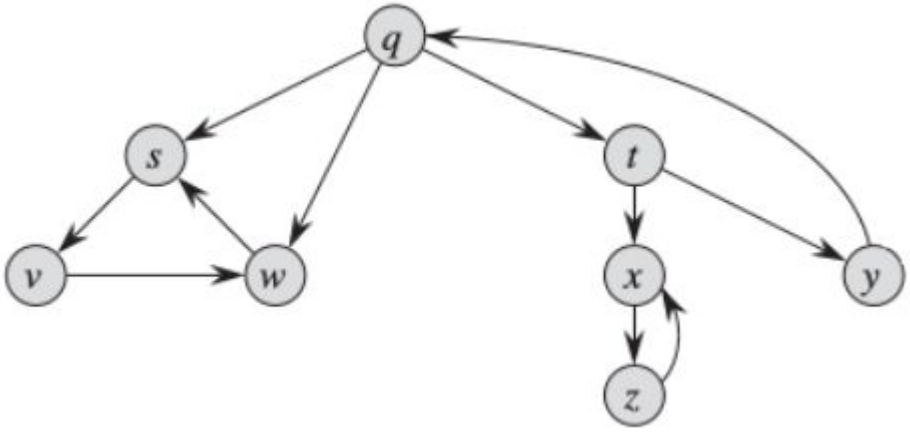
(a) [**5 points**]What is the max-product-sum of [1,4,3,2,3,4,2]?

(b) **[15 points]** Write the recurrence relation and a pseudo-code for max-product-sum problem.

3. **[20 points]** There are n gas stations $s_1, s_2, s_3, \ldots, s_n$ along the highway. On a full tank of gasoline, your car goes for D miles. Gas station s_i, for $2 \leq i \leq n$, is d_i miles after the s_{i-1}, and each s_1, s_n locates at the start and the end of the highway. Write a pseudo algorithm that finds the minimum number of the gas stations you have to visit until you exit the highway and analyze the time complexity of your algorithm. (Suppose that you filled up your car before departing) .

4. **[20 Points]** Answer the following questions about BFS and DFS.

(a) **[10 Points]** Draw the tree that can be obtained by running BFS and DFS on the graph below. Assume both algorithms start from the node q.



(b) **[10 Points]** Again, suppose we run BFS and DFS on another undirected graph G', both start from the same node, and we find that the trees are the same at this time. Does this necessarily mean that G' is a tree? Justify your answers.