

ads-yang2020-project-list

P1: Shortest Path Algorithm with Heaps (cls-5)

- This project requires you to implement Dijkstra's algorithm based on a min-priority queue, such as a **Fibonacci heap**.
- The goal of the project is to find the best data structure for Dijkstra's algorithm.

P2: Safe Fruit (cls-6)

- There are a lot of tips telling us that some fruits must not be eaten with some other fruits, or we might get ourselves in serious trouble.
- For example, bananas can not be eaten with cantaloupe (哈密瓜), otherwise it will lead to kidney deficiency (肾虚).
- Now you are given a long list of such tips, and a big basket of fruits. You are supposed to pick up those fruits so that it is safe to eat any of them.

P3: Beautiful Subsequence (cls-8)

- Given an integer m , we consider a sequence (with at least 2 elements) as **beautiful** if it contains 2 neighbors with difference no larger than m .
- Given an integer sequence with n elements, your job is to calculate the number of beautiful subsequences in it.

P4: Huffman Codes (cls-9)

- In 1953, David A. Huffman published his paper "**A Method for the Construction of Minimum-Redundancy Codes**", and hence printed his name in the history of computer science.
- As a professor who gives the final exam problem on Huffman codes, I am encountering a big problem: the Huffman codes are NOT unique.
- The students are submitting all kinds of codes, and I need a computer program to help me determine which ones are correct and which ones are not.

P5: Texture Packing (cls-11)

- Texture Packing is to pack multiple rectangle shaped textures into one large texture. The resulting texture must have a given width and a **minimum** height.
- You are to design and analyze an approximation algorithm that runs in polynomial time.

P6: Skip Lists (cls-13)

- **Skip list** is a data structure that supports both searching and insertion in $O(\log(N))$ expected time.
- This project requires you to introduce the skip lists, and to implement insertion, deletion, and searching in skip lists.

P7: MapReduce (cls-14)

- **MapReduce** is a programming model and an associated implementation for processing and generating large data sets with a parallel, distributed algorithm on a cluster. A MapReduce program is composed of a **Map()** procedure and a **Reduce()** procedure.
- In this project, you are supposed to briefly introduce the framework of MapReduce, and implement a MapReduce program to count the appearance of each word in a set of documents.