Frequent asked Interview Algorithms

From Here

- [ ] Graph:
  - [x] ~~BFS 没啥好说的，跳过~~
    - Complexity: `O(V+E)`
  - [x] ~~DFS：注意：Unlike trees, graphs may contain cycles, so we may come to the same node again. To avoid processing a node more than once, we use a boolean visited array. 其实也没啥好说的。~~
    - Complexity: `O(V+E)`
  - [x] ~~Dijkstra: Shortest path to a particular target~~
    - Complexity: `O(Elog(V))` with the help from binary heap

```
// Pick the minimum distance vertex from the set of vertices not
// yet processed. u is always equal to src in the first iteration.
int u = minDistance(dist, sptSet);
```

Not efficient

  - [x] ~~Shortest Path from every points to every other points: Floyd Warshall~~
    - Complexity: $O(V^3)$

```
for(int k=0; k<n; ++k) for(int i=0; i<n; ++i) for(int j=i+1; j<n; ++j) path[i][j] = min(path[i][j], path[i][k] + path[k][j]);
Remember to put the intermediate node index k at the outest loop!!!!
```

  - [x] ~~Detect Cycle in a Graph: Union Find 没啥好说的。~~
    - Complexity: `O(E+V)`
  - [x] ~~Minimum Span Tree: Prim 感觉跟下面那个没啥区别！~~
    - Complexity: `O(Elog(V))`
    - At every step, it considers all the edges that connect the two sets, and picks the minimum weight edge from these edges. After picking the edge, it moves the other endpoint of the edge to the set containing MST.
  - [x] ~~Minimum Span Tree: Kruskal 本科学过的算法，浅显易懂！~~
    - Complexity: `O(Elog(E))`

    ***1.** Sort all the edges in non-decreasing order of their weight.*

2. *Pick the smallest edge. Check if it forms a cycle with the spanning tree formed so far. If cycle is not formed, include this edge. Else, discard it.*
3. *Repeat step#2 until there are (V-1) edges in the spanning tree.*

- [ ] Topological sort
  - [x] ~~更新入度法:~~
    - Complexity: `O(E + V)`
  - [ ] DFS @Zebo L
- [x] ~~Boggle: Find all possible words in a board of characters.~~
  - The idea is to consider every character as a starting character and find all words starting with it. All words starting from a character can be found using Depth First Traversal. We do depth first traversal starting from every cell. We keep track of visited cells to make sure that a cell is considered only once in a word. 四五没有什么新东西。
- [ ] Bridge in a Graph
  - Define Bridge in a Graph:
    - An edge in an undirected connected graph is a bridge iff removing it disconnects the graph. For a disconnected undirected graph, definition is similar, a bridge is an edge removing which increases number of disconnected components.

- [x] ~~Linked List~~
  - [x] ~~Insert: 没啥好说的~~
    - Complexity: `O(n)`
  - [x] ~~Delete: 没啥好说的~~
    - Complexity: `O(n)`
  - [x] ~~Compare two string~~
  - [x] ~~Add Two numbers~~
  - [x] ~~Alternative Merge~~
  - [x] ~~Merge Sort~~
  - [x] ~~Reverse~~
  - [x] ~~Union and Intersection~~
  - [x] ~~Detect and Remove Loop~~
  - [x] ~~Merge Sort: 快慢指针~~
  - [x] ~~Select a random node from a linked list (single linked)~~
    - One pass:

```
    current = head->next;
```

```
    results = head->key;
    for (n=2; current!=NULL; n++){
        // change result with probability 1/n
        if (rand() % n == 0)
           result = current->key;
        // Move to next node
        current = current->next;
    }
```

- The probability that a particular node is chosen is $1/j * j/(j+1) * ... * (N-1)/N$

- [ ] Reservoir sampling:
  - Create an array *reservoir[0..k-1]* and copy first *k* items of *stream[]* to it.
  - Now one by one consider all items from *(k+1)*th item to *n*th item.
    - Generate a random number from 0 to *i* where *i* is index of current item in *stream[]*. Let the generated random number is *j*.
    - If *j* is in range 0 to *k-1*, replace *reservoir[j]* with *arr[i]*

- [x] ~~Dynamic Programming: GeeksForGeeks 给的题都不难~~

- [x] ~~Tree:~~
  - [x] ~~Subtree Checking~~
    - [x] ~~Preorder + inorder:~~
      - **1**) Find inorder and preorder traversals of T, store them in two auxiliary arrays inT[] and preT[].
      - **2**) Find inorder and preorder traversals of S, store them in two auxiliary arrays inS[] and preS[].
      - **3**) If inS[] is a subarray of inT[] and preS[] is a subarray preT[], then S is a subtree of T. Else not.
    - [x] ~~DFS~~
    - [x] ~~Serialization~~

- [x] ~~Sort:~~
  - [x] ~~Merge Sort~~
  - [x] ~~Heap Sort~~
  - [x] ~~Quick Sort~~

- [x] ~~Array and String:~~
  - [x] ~~Pythagorean Triplets ($a^2 + b^2 = c^2$)~~
    - 这个先sort是好方法！
  - [x] ~~Find the smallest integer that cannot be represented as sum of any subset of an array.~~
    - 思路是dp，每次往大扩张 `a[i]`
  - [x] ~~Smallest subarray sum with value greater than a threshold:~~
    - Partial sum + sliding window

- [ ] Bit Manipulation
  - [x] ~~Max XOR sum Subarray:~~
    - Build a tier for partial XOR sum
    - For each partial XOR sum, use the tier to find another partial XOR sum so that there XOR is minimized.