Q1
get start vertex
visited = true;
while start!=NULL
    if not visited
        dfs
    start = start->next;

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

SC1007 Data Structures and Algorithms **2022/23 Semester 2**

Tutorial 6: Backtracking and Dynamic Programming

School of Computer Science and Engineering *Nanyang Technological University*

Q1 Give a pseudocode of finding a simple path connecting two given vertices in an undirected graph by Depth-First-Search.

Q2 Design a backtracking algorithm to print out all possible permutation of a given sequence. For example, input is given as "1234". The 24 output permutations are printed out from "1234" to " 4321".

Q3 Find length of longest substring of a given string of digits, such that sum of digits in the first half and second half of the substring is same. For example, if the input string is "142124", the whole string is the answer. The sum of the first 3 digits = the sum of the last 3 digits (1+4+2) = 1+2+4. Thus, the length is 6. If the input is "12345678", then the output is 0. If the input is "9430723", then the output is 4 (4307).

Q2
-create permute() with parameters as input string, starting index of string, ending index of string
-call this function with values input string, 0, size of string -1
    -in the function, if L==r, print the same string
    -else run a for loop from L to r and swap the current element in the for loop with the inputString[L]
    -call the function again but with L+1
    -swap the previously swapped values to initiate backtracking

```
void swap(char* x,char* y){
  char temp;
  temp = *x;
  *x = *y;
  *y = temp;
}

void permute(char*a, int l, int r){
   int i;
   if(L==r)
     print a;
   else{
       for(i = L;I<=r;i++){
           swap((a+L),(a+i));
           permute(a,L+1,r);
           swap((a+L),(a+i));

int main{
    ...
   permute(str,0,n-1);
}
```

O(n!)

```
int find length(string str, int n){

    int sum[n+1];
    sum[0] = 0;

    for(i = 1;i<=n;i++){
        sum[i] = (sum[i+1] +  str[i-1] - '0');      //sum of prev sum[i] value and current value

    int ans = 0;

    for(len = 2;len<=n;len+=2){
       for(i = 0; i<=n-len;i++){

          //int j = i+ len -1;

          if(sum[i+len/2] - sum[i] == sum[i+len] - sum[i+len/2])
              ans = max(ans,len);

    return ans;
```