### Manacher

vector<int> radius, let;  
string expa\_str;  
void Manacher(const string &str, int &pos, int &max\_len) {  
 int orig\_len = str.length();  
 int expa\_len = (orig\_len + 1) << 1;  
 max\_len = 0;  
 radius.resize(expa\_len + 1);  
 expa\_str.resize(expa\_len + 1);  
 //@#0#1#2#3#4#5#6#7#8#9#$  
 expa\_str[0] = '@';  
 expa\_str[1] = '#';  
 expa\_str[expa\_len] = '$';  
 for (int i = 1; i <= orig\_len; ++i) {  
 expa\_str[i << 1] = str[i-1];  
 expa\_str[i << 1 | 1] = '#';  
 }  
 radius[1] = 1;  
 for (int max\_R = 0, center = 0, i = 2; i < expa\_len; ++i) {  
 radius[i] = i < max\_R ? min(max\_R-i, radius[2\*center-i]) : 1;  
 while (expa\_str[i-radius[i]] == expa\_str[i+radius[i]])  
 ++radius[i];  
 if (radius[i] + i > max\_R) {  
 max\_R = radius[i] + i;  
 center = i;  
 }  
 if (radius[i]-1 > max\_len) {  
 max\_len = radius[i]-1;  
 pos = (center - radius[i] + 1) << 1;  
 }  
 }  
}  
  
//odd为false，字符串为奇回文串  
int start\_mid(int x, bool odd) {  
 return odd ? radius[(x+1) << 1] - 1 : radius[(x+1) << 1 | 1] - 1;  
}  
  
//知道回文左边界，且在Manacher函数运行结束后使用  
int start\_left(int x, string str) {  
 int expand\_len = (str.length() + 1) << 1;  
 let.resize(expand\_len + 1);  
 for (int i = 0; i <= expand\_len; i++)  
 let[i] = 0;  
 for (int i = 2; i < expand\_len; i++)  
 if (let[i - radius[i] + 1] < i + 1)  
 let[i - radius[i] + 1] = i + 1;  
 for (int i = 1; i <= expand\_len; i++)  
 if (let[i] < let[i - 1])  
 let[i] = let[i - 1];  
 return let[(x + 1) << 1] - ((x + 1) << 1);  
}