

2112233062—何智鹏-第二次作业

1. 假设Kmp算法当输入为S=ABDABABAC，w=ABAC，请写出Kmp算法的执行过程
2. 写出Kmp搜索的完整伪代码，并试着分析动态规划思想在Kmp算法中是怎样体现的（关于dp的体现：通过...写出0/1背包问题的最优解构造算法

2022年11月17日

1. 假设Kmp算法当输入为S=ABDABABAC，w=ABAC，请写出Kmp算法的执行过程

前缀表next[]

A	B	A	C
0	0	1	0

第一次循环开始： ABDABABAC
ABAC

第一次循环停止： AB**D**ABABAC
AB**A**C

next[j] = 0，从i位置从头开始匹配

第二次循环开始： ABDABABAC
ABAC

第二次循环停止： AB**D**ABABAC
ABAC

next[j] = 0，从i位置从头开始匹配

第三次循环开始： ABDABABAC
ABAC

第三次循环停止： ABDABAB**B**AC
ABAC**C**

next[j] = 1; 从i位置开始匹配第二个

第四次循环开始: ABDABABAC
 ABAC

第四次循环停止: ABDABABAC
 ABAC

匹配结束。

2. 写出Kmp搜索的完整伪代码，并试着分析动态规划思想在Kmp算法中是怎样体现的（关于dp的体现：通过存储 `T[i]`，我们可以快速丢弃某些子问题）。

- 使用 `next` 数组作匹配，`i` 指向文本串的起始位置，`j` 指向模式串的起始位置
 - 如果 `s[i] != s[j + 1]`，则 `j` 就要从 `next` 数组中寻找下一个匹配的位置；
 - 如果 `s[i] == s[j + 1]`，那么 `i` 和 `j` 同时向后移动；
 - 如果 `j` 指向了模式串的末尾，那么说明完全匹配了

```
1 for (int i = 0; i < s.size(); i++) {  
2     while (j && s[i] != t[j + 1]) j = next[j - 1];  
3     if (s[i] == t[j + 1]) j++; //i的增加逻辑在for循环中  
4  
5     if(j == (t.size() - 1) return i - t.size() + 1;  
6 }
```

C++ | 复制代码

写出0/1背包问题的最优解构造算法

```
1  #include<bits/stdc++.h>
2
3  using namespace std;
4
5  const int MAXN = 1005;
6  int v[MAXN];    // 体积
7  int w[MAXN];    // 价值
8  int f[MAXN][MAXN]; // f[i][j], j体积下前i个物品的最大价值
9
10 int main()
11 {
12     int n, m;
13     cin >> n >> m;
14     for(int i = 1; i <= n; i++)
15         cin >> v[i] >> w[i];
16
17     for(int i = 1; i <= n; i++)
18         for(int j = 1; j <= m; j++)
19         {
20             // 当前背包容量装不进第i个物品, 则价值等于前i-1个物品
21             if(j < v[i])
22                 f[i][j] = f[i - 1][j];
23             // 能装, 需进行决策是否选择第i个物品
24             else
25                 f[i][j] = max(f[i - 1][j], f[i - 1][j - v[i]] + w[i]);
26         }
27
28     cout << f[n][m] << endl;
29
30     return 0;
31 }
```