ACM-template

langman

December 23, 2017

1 head.set

```
#include <iostream >
     #include < cstdio >
     #include < cstring >
     #include < string >
     #include < algorithm >
6
     #include < queue >
     #include < stack >
     #include < vector >
9
     #include < cmath >
10
     #include < set >
11
     #include < cstdlib >
12
     #include < functional >
13
     #include < climits >
     #include < cctype >
14
     #include < iomanip >
15
16
     using namespace std;
17
     typedef long long 11;
     #define INF 0x3f3f3f3f
18
     const int mod = 1e9+7;
19
20
     #define clr(a,x) memset(a,x,sizeof(a))
21
     #define cle(a,n) for(int i=1;i<=n;i++)
22
     a.clear();
23
     const double eps = 1e-6;
24
     int main()
25
     freopen("in.txt", "r", stdin);
26
     freopen("out.txt","w",stdout);
27
28
29
     return 0;
```

2 DP

2.1 LIS/LDS

最大上升子序列,下降,严格上,严格降. nlogn的复杂度,调用库里面的函数

```
2
   LIS (LDS)
   template < class Cmp>
   int LIS (Cmp cmp) (nlogn)
5
        static int m, end [N];
6
 7
       m = 0;
8
        for (int i = 0; i < n; i ++)</pre>
9
          int pos=lower_bound(end, end+m, a[i],cmp)-end;
10
          end[pos]=a[i],m+=pos==m;
11
12
        }
13
        return m;
14 }
        cout << LIS(less<int>()) << endl;</pre>
15
        cout << LIS(less_equal<int>()) << endl;</pre>
16
        cout << LIS(greater<int>()) << endl;</pre>
17
        cout << LIS(greater_equal<int>()) << endl;</pre>
18
```

2.2 dp in bag

背包的题目,是比较基础的dp类型的题 01和完全背包相对来说比较简单,n方的复杂度 部分背包的话相对来说,因该是把部分背包换成01背包,降低复杂度

他的状态点在于 当背包容量为x时,他的最佳状态 然后找出容量是x的时候能从 哪几个子状态转移过来。 难点在于:

- 1: 背包的构造
- 2: 背包状态转移方程的寻找
- 3:方向是从前到后,还是从后到前
- 4: dp维数的确定

```
for(int i = 0; i < num; i++)</pre>
      for(int j = v; j \ge money[i]; j++)
 5
         dp[j] = max(dp[j], dp[j-money[i]]+value[i]);
 7
       }
8
   }
10
   // part bag (better make it into 0 1 bag)
11
12
   //full \ bag
13 for(int i = 0; i < num; i++)
14
     for(int j = v; j \ge money[i]; j --)
15
16
17
        dp[j] = max(dp[j],dp[j-money[i]]+value[i]);
18
19
   }
```

2.3 dp in tree