数据结构与算法1 作业12

2019201409 于倬浩

15.2-5

显然可以直接根据DP计算过程，先枚举计算的区间，再枚举每次将区间切成两半的方案数，再乘以每次分出的区间数目2即可。

$\sum_{i=1}^{n}\sum_{j=1}^{n}R(i,j)=\sum_{l=2}^{n}(n-l+1)\times(l-1)\times 2$

$=2(n+2)\sum_{l=2}^{n}l-2(n+1)(n-2+1)-2\sum_{l=2}^{n}l^2$

化简可得，$ans=\frac{n^3-n}{3}$。

15.3-6

当c_i均为0时，如果按照兑换次数划分阶段，维护`f[i]`表示经过若干次兑换，从1兑换到i的最优汇率，那么此时显然具有无后效性和最优子结构。考虑当前从`1`兑换到`i`的最优解，假设中间某一步是由`1`经过若干次兑换到`j`，再从`j`经过一步兑换到`i`。如果从`1~j`的兑换过程存在更优的解法，那么在计算当前最优解时，显然会把更优的解法计入`f[j]`，因为直接替换解法并不会对之后的计算过程产生影响，所以可以直接换成更优秀的。

但是，当c_i不为0时，如果依旧只维护一维`f[i]`，那么显然不能做到无后效性，也就无法保证最优子结构。此时操作次数和兑换代价挂钩。即有可能出现某种解法，在兑换过程中的某一些步骤汇率不为最优，但由于兑换次数较少，手续费较少导致最终总代价较小。然而，对于这种情况其实也有办法解决。只需修改状态的定义，设`f[i][j]`表示经过了`j`次兑换，从`1`兑换到`i`的最优代价，只需按照兑换次数`j`划分阶段，每次计算出所有`i`的值即可。最后计算答案时，再枚举`f[n][j]*d-`

sum_c[j]即可，这样可以保证，对于指定兑换次数的操作序列满足最优子结构和无后效性，因此修改后的算法具有正确性。