

JAG Summer Camp Day4 I: Live Programming

原案：岸本

問題文：矢藤

解答：岸本・矢藤・澤

解説：岸本

問題概要

- ▶ 合計 T 分の間歌を歌う
- ▶ 同じ曲は一曲しか歌わない
- ▶ 歌える曲は N 曲あり、それぞれの曲 i に対してパラメータが定まっている
 - ▶ 歌の長さ $t[i]$
 - ▶ 歌うことで得られる満足度を示す整数 $p[i]$
 - ▶ その曲の特徴を表す整数 $f[i]$
- ▶ カラオケ全体を通しての満足度とは、歌った曲の満足度の合計から、「 j 番目に歌った曲と $j+1$ 番目の曲の特徴の差 $(f[j+1]-f[j])^2$ 」の合計を引いたものである
- ▶ T 分を越えて歌えない制約の中で、カラオケ全体を通しての満足度を最大化せよ

考察

わかること

- ▶ ナップサック問題の類型である
- ▶ 最適解での曲順は f の昇順(or 降順)になる

なので、

f を昇順ソートしてからナップサックっぽいDPを構築していこう

DP表を作ろう

- ▶ 保存する情報
DP[i][j] := i番目の曲までを歌ってjの時間を使ったときの最大満足度
- ▶ 初期化
DP[a][t[a]] = p[a]
DP[i][j] = -INF (other)
- ▶ 更新
DP[i][j] = (max_{1<=k<=i-1} DP[k][j-t[i]] + p[i] - (f[i]-f[k])^2)

この表をナイーブに更新すると $O(TN^2)$ かかる。

困った。

DP表を作ろう

- ▶ 保存する情報

$DP[i][j] := i$ 番目の曲までを歌って j の時間を使ったときの最大満足度

- ▶ 初期化

$DP[a][t[a]] = p[a]$

$DP[i][j] = -INF$ (other)

- ▶ 更新

$DP[i][j] = (\max_{1 \leq k \leq i-1} DP[k][j-t[i]] + p[i] - (f[i]-f[k])^2)$

$= (\max_{1 \leq k \leq i-1} (DP[k][j-t[i]] - f[k]^2) + 2f[i]f[k]) + p[i] - f[i]^2$

変形した。

この更新は、蟻本のdequeの問題例(K-Anonymous Sequence)と同じ型をしている

よって、そこを参考にすれば時間計算量 $O(TN)$ で解ける

そのほか

▶ なぜ高速に解けるとわかるのか

- ▶ こういう問題はkに依存しない箇所を整理すると見通しが良くなることがある
 - ▶ $p := j - t[i]$, $x := f[i]$, $C := p[i] - f[i]^2$ とする(i, j を固定すればこれらは定数)
 - ▶ $DP[k][p] - f[k]^2 + 2f[k]x + C$ となる
 - ▶ $a_k := 2f[k]$, $b_k := DP[k][p] - f[k]^2$ とする
 - ▶ すると x で $a_k x + b_k$ を最小化する k を求める問題になる
 - ▶ これはエンベロープ系の問題なので高速に解けるとわかる
- ▶ 直観的観察だけでなく、式を落ち着いて観察することも大事

▶ エンベロープ系は大変そう

- ▶ 実は f はソートされているので $a_k = f[k]$ は昇順に出てくるので比較的楽
 - ▶ よって、蟻本の問題とよく似ているという話になります

▶ INFはある程度大きくしないとWAするので気を付けてください

統計情報

- ▶ FA : Kut_Pioneer (231:27)
- ▶ AC/Try : 2/3
- ▶ AC/Submission : 2/10

- ▶ Writer解
 - ▶ 岸本 : C++ 107行
 - ▶ 矢藤 : C++ 97行
 - ▶ 矢藤 : Java 118行
 - ▶ 澤 : C++ 69行