

J : Farm Village

原案 : tokoharu

実装 : tokoharu

not

概要

- 一直線上にN個の家がある
- それぞれの家では2単位の穀物を育てることができる
- それぞれの家で1単位の穀物がほしい
- 家 i での1単位あたりの穀物を育てるコスト p_i と,
家 $i, (i+1)$ 間の1単位当たりの穀物に対する移動コスト d_i , が与えられる
- 条件を満たす生産方法のうち、全体の最小コストはいくらか？

- $N \leq 10^5$

解法(1)

- 最小費用流で解けるが、頂点が大きすぎるのでだめ。
- そこで、「(i, i+1)間で流れるフローの量」を基準にDPを立ててみます
- $DP[i][j] := \min_{k=j-1, j, j+1} DP[i-1][k] + (k-j+1) p_i + |j| d_i$
- $DP[N][0]$ を求めたい
- これは最悪 $O(N^2)$ かかるので改善したい。

解法(2)

- 着眼点：凸性
 - $f_i(x) = dp[i][x]$ を端点とする区分解線形関数を考える
 - 初期関数は凸。($f_0(0) = 0, f_0(x) = \inf$)
 - 絶対値関数は凸なので足しても凸。
 - 周辺の3要素のminをとっても凸。
- だから $f_i(x) = dp[i][x]$ は凸関数とみなせる
- この関数は折れ線関数なので、折れ線の傾きを適宜管理すればよい。

解法(3)

- 管理方法はいろいろあると思う
- 想定解法ではpriority_queueをふたつ用いた
 - これは絶対値関数を加算をやりやすくする目的。
 - それぞれ正側の傾きと負側の傾きを持つ。
 - 当然、これらは加算で変化するが、その変化分も変数として持つておく。
 - min関数をとった時に、正側で管理していた傾きが負側へ移動することがある
 - しかしこれは高々1個分なので、片方のpriority_queueからpopして、もう片方へpushするようなことをすればよい
- 結局 $O(N \log N)$ で計算可能である

notさんのコード

```
int main() {
    int64_t n, a = 0;
    cin >> n;
    vector<int64_t> d(n - 1), p(n);
    for (auto& i : d) cin >> i;
    for (auto& i : p) cin >> i;
    priority_queue<int64_t> p1;
    priority_queue<int64_t, vector<int64_t>, greater<int64_t>> p2;
    p1.emplace(p[0]);
    p2.emplace(p[0]);
    int64_t res = 0;
    for (int i = 0; i < n - 1; ++i) {
        res += (i + 1) * d[i];
        a += d[i];
        if (p[i + 1] < p2.top() + a) {
            p1.emplace(p[i + 1] + a);
            p1.emplace(p[i + 1] + a);
            p2.emplace(p1.top() - 2 * a);
            p1.pop();
        } else {
            p2.emplace(p[i + 1] - a);
            p2.emplace(p[i + 1] - a);
            p1.emplace(p2.top() + 2 * a);
            p2.pop();
        }
    }
    while (!p1.empty()) {
        auto i = p1.top();
        p1.pop();
        res += i - a;
    }
    cout << res << endl;
}
```

裏の問題

- 実はこの問題の双対は、ドワンゴプロコンの『花火』と似たような問題
 - 花火は、数直線上でプラス方向に任意に動け、不満度が花火ごとに $|x - p_i|$ で与えられる、
 - 不満度を最小化せよ。
- 今回の問題の双対問題は次。
 - 花火を見たいが、単位時間あたり-1,0,1にしか移動できないので、不満度を最小化せよ
- 双対側の問題も主側の問題の解法と同じようなもので解ける。

ジャッジ解

not C++ 43 lines, 928B

tokoharu C++ 73 lines, 1.4kB

hankan_rta 111:47

CxivDxiv 158:03