DP优化 学习小结

2010-08-27 15:55

|  |
| --- |
| 这两天简单学习了一些动态规划的优化技巧，感觉自己的数学实在太弱了，虽然大概知道怎么用，但很多证明自己理解的还不是很透。 关于用**单调队列优化**的动态规划以前整理过一些，可以看这里：【[单调队列 学习小结](http://hi.baidu.com/%B1%BF%D0%A1%BA%A2_shw/blog/item/2910950e9147ce276159f34d.html)】 **四边形不等式优化** 主要是对于dp（i，j）的决策点 s（i，j），通过单调性证明 s（i， j - 1）<= s（i，j） <= s（i + 1，j） 推荐2道题目： **hdu3480  【** [**Division**](http://acm.hdu.edu.cn/showproblem.php?pid=3480)**】 hdu2829  【** [**Lawrence**](http://acm.hdu.edu.cn/showproblem.php?pid=2829)**】**  2829 的代码：  int n, m, num[1005], s[1005][1005]; long long w[1005][1005], dp[1005][1005]; int main(int argc, char\*\* argv) {  **while** (scanf("%d%d", &n, &m) != EOF) {  **if** (n == 0 && m == 0) **break**;  **for** (int i = 1; i <= n; i++) num[i] = next\_int();  **for** (int i = 1; i <= n; i++) {  int t = 0;  w[i][i - 1] = 0;  FF(j, i, n) w[i][j] = w[i][j - 1] + t \* num[j], t += num[j];  }  **if** (m == 0) {  printf("%d**\n**", w[1][n]);  **continue**;  }  **for** (int i = 1; i <= n; i++) dp[i][0] = w[1][i], s[i][0] = 0;  **for** (int j = 1; j <= m; j++) {  s[n + 1][j] = n - 1;  **for** (int i = n; i >= j + 1; i--) {  int head = max(s[i][j - 1], j);  int rear = min(s[i + 1][j], i - 1);  dp[i][j] = max\_long;  **for** (int k = head; k <= rear; k++) {  int temp = dp[k][j - 1] + w[k + 1][i];  **if** (temp < dp[i][j]) dp[i][j] = temp, s[i][j] = k;  }  }  }  cout << dp[n][m] << endl;  }   **return** (EXIT\_SUCCESS); } **斜率优化** 推荐大家去看这篇论文【[浅谈数形结合思想在信息学竞赛中的应用](http://wenku.baidu.com/view/f6421ce8b8f67c1cfad6b8ac.html) 】 对于这类题目： 首先，证明斜率满足单调性，对于每个点找到其x 和 y。 然后，用一个队列，队首维护每次的最优决策点，队尾进队时维护一个下凸的折线。 推荐几个题目： **hdu2933  【** [**Fairies' Defence**](http://acm.hdu.edu.cn/showproblem.php?pid=2933)**】** 就是论文中的原题 **apio2010  【**[特别行动队](http://61.187.179.132:8080/JudgeOnline/showproblem?problem_id=1911)**】**  #define maxn 1000010 **struct** point {  double x, y; }po[maxn]; double sum[maxn], dp[maxn], a, b, c; int n, arr[maxn]; **inline** double cal(double x) {  **return** a \* x \* x + b \* x + c; } **inline** double rate(int p1, int p2) {  **return** (po[p1].y - po[p2].y) / (po[p1].x - po[p2].x); } int main(int argc, char\*\* argv) {  scanf("%d", &n);  scanf("%lf%lf%lf", &a, &b, &c);  sum[0] = 0.0;  int t;  **for** (int i = 1; i <= n; i++) {  t = next\_int();  sum[i] = sum[i - 1] + t;  }  po[0].x = 0.0;  po[0].y = 0.0;  int head = 0, rear = 0;  **for** (int i = 1; i <= n; i++) {  **if** (head > rear) head = rear;  **for** (int j = head; j <= rear - 1; j++) {  **if** (rate(arr[head], arr[head + 1]) <= 2 \* sum[i]) head += 1;  **else** **break**;  }  dp[i] = dp[arr[head]] + cal(sum[i] - sum[arr[head]]);  po[i].x = a \* sum[i];  po[i].y = dp[i] + a \* sum[i] \* sum[i] - b \* sum[i];  arr[++rear] = i;  **while** (rear >= 2) {  **if** (rate(arr[rear - 2], arr[rear - 1]) >= rate(arr[rear - 1], arr[rear])) {  arr[rear - 1] = arr[rear];  rear -= 1;  } **else** **break**;  }  }  cout << (long long)dp[n] << endl;   **return** (EXIT\_SUCCESS); } **hnoi2008  【**[玩具装箱toy](http://61.187.179.132:8080/JudgeOnline/showproblem?problem_id=1010)**】**  #define maxn 50010 **struct** point {  double x, y; }po[maxn]; double sum[maxn], f[maxn], dp[maxn], c; int n, arr[maxn]; **inline** double rate(int p1, int p2) {  **return** (po[p1].y - po[p2].y) / (po[p1].x - po[p2].x); } **inline** double cal(int p1, int p2) {  **return** dp[p2] + (f[p1] - f[p2] - c) \* (f[p1] - f[p2] - c); } int main(int argc, char\*\* argv) {  scanf("%d%lf", &n, &c);  c += 1.0;  int t;  sum[0] = 0;  f[0] = 0;  **for** (int i = 1; i <= n; i++) {  t = next\_int();  sum[i] = sum[i - 1] + t;  f[i] = sum[i] + i;  }  po[0].x = 0;  po[0].y = c \* c;  dp[0] = 0.0;  int head = 0, rear = 0;  **for** (int i = 1; i <= n; i++) {  **if** (head > rear) head = rear;  **for** (int j = head + 1; j <= rear; j++) {  **if** (cal(i, arr[j]) <= cal(i, arr[head])) head += 1;  **else** **break**;  }  dp[i] = dp[arr[head]] + (f[i] - f[arr[head]] - c) \* (f[i] - f[arr[head]] - c);  po[i].x = 2 \* f[i];  po[i].y = dp[i] + (f[i] + c) \* (f[i] + c);  arr[++rear] = i;  **while** (rear >= 2) {  **if** (rate(arr[rear - 2], arr[rear - 1]) >= rate(arr[rear - 1], arr[rear])) {  arr[rear - 1] = arr[rear];  rear -= 1;  } **else** **break**;  }  }  cout << (long long)dp[n] << endl;   **return** (EXIT\_SUCCESS); } **hdu3045  【** [**Picnic Cows**](http://acm.hdu.edu.cn/showproblem.php?pid=3045)**】 && pku3709  【**[**K-Anonymous Sequence**](http://162.105.81.212/JudgeOnline/problem?id=3709)**】** 这题的数据范围貌似比较大，直接用实数做不行，用long long 才行。  #define maxn 400100 long long num[maxn], sum[maxn], dp[maxn]; int n, m, arr[maxn]; **inline** long long cal(int p1, int p2) {  **return** dp[p2] + sum[p1] - sum[p2] - num[p2 + 1] \* (p1 - p2); } **inline** long long SS(int p1, int p2) {  **return** num[p1 + 1] - num[p2 + 1]; } **inline** long long GG(int p1, int p2) {  **return** dp[p1] - sum[p1] + p1 \* num[p1 + 1] - (dp[p2] - sum[p2] + p2 \* num[p2 + 1]); } **inline** bool rate(int p1, int p2, int p3) {  **if** (GG(p1, p2) \* SS(p2, p3) >= GG(p2, p3) \* SS(p1, p2)) **return** **false**;  **else** **return** **true**; } int main(int argc, char\*\* argv) {  **while** (scanf("%d%d", &n, &m) != EOF) {  **for** (int i = 1; i <= n; i++) num[i] = next\_int();  sort(num + 1, num + n + 1);  sum[0] = 0;  **for** (int i = 1; i <= n; i++) sum[i] = sum[i - 1] + num[i];  dp[0] = 0;  int head = 0, rear = 0;  arr[++rear] = 0;  **for** (int i = m; i <= n; i++) {  **for** (int k = head + 1; k <= rear; k++) {  **if** (cal(i, arr[k]) <= cal(i, arr[head])) head += 1;  **else** **break**;  }  dp[i] = cal(i, arr[head]);  **if** (i - m + 1 >= m) arr[++rear] = i - m + 1;  **while** (head + 2 <= rear) {  **if** (rate(arr[rear - 2], arr[rear - 1], arr[rear]) == **false**) {  arr[rear - 1] = arr[rear];  rear -= 1;  } **else** **break**;  }  }  cout << dp[n] << endl;  }   **return** (EXIT\_SUCCESS); } |