Solutions For Claris' Contest # 2 Day 2

Claris

Hangzhou Dianzi University

2016 年 9 月 4 日

Divisors(div.c/cpp/pas)

给定 m 个不同的正整数 $a_1, a_2, ..., a_m$ ，请对 0 到 m 每一个 k 计算，在区间 $[1, n]$ 里有多少正整数是 a 中恰好 k 个数的约数。

Divisors(div.c/cpp/pas)

给定 m 个不同的正整数 $a_1, a_2, ..., a_m$，请对 0 到 m 每一个 k 计算，在区间 $[1, n]$ 里有多少正整数是 a 中恰好 k 个数的约数。

- 对于 30% 的数据，$m \le 200$，$n, a_i \le 1000$。

Divisors(div.c/cpp/pas)

给定 m 个不同的正整数 $a_1, a_2, \ldots, a_m$，请对 0 到 m 每一个 k 计算，在区间 $[1, n]$ 里有多少正整数是 a 中恰好 k 个数的约数。

- 对于 30% 的数据，$m \le 200$，$n, a_i \le 1000$。
- 对于另外 30% 的数据，$m = 1$，$n, a_i \le 10^9$。

Divisors(div.c/cpp/pas)

给定 m 个不同的正整数 $a_1, a_2, ..., a_m$，请对 0 到 m 每一个 k 计算，在区间 $[1, n]$ 里有多少正整数是 a 中恰好 k 个数的约数。

- 对于 30% 的数据，$m \le 200$，$n, a_i \le 1000$。
- 对于另外 30% 的数据，$m = 1$，$n, a_i \le 10^9$。
- 对于 100% 的数据，$m \le 200$，$n, a_i \le 10^9$。

30 分做法 1

- 对于 30% 的数据，$m \le 200$，$n, a_i \le 1000$。

30 分做法 1

- 对于 30% 的数据，$m \le 200$，$n, a_i \le 1000$。
- 暴力枚举 1 到 n 里每个数，统计它是多少个数的约数即可。

30 分做法 1

- 对于 30% 的数据，$m \leq 200$，$n, a_i \leq 1000$。
- 暴力枚举 1 到 n 里每个数，统计它是多少个数的约数即可。
- 时间复杂度 $O(nm)$。

30 分做法 2

- 对于另外 30% 的数据，$m = 1$，$n, a_i \leq 10^9$。

30 分做法 2

- 对于另外 30% 的数据，$m = 1$，$n, a_i \leq 10^9$。
- 对于 $k = 1$ 的答案，枚举 a_1 的全部约数即可，剩下的就是 $k = 0$ 的答案。

30 分做法 2

- 对于另外 30% 的数据，$m = 1$，$n, a_i \le 10^9$。
- 对于 $k = 1$ 的答案，枚举 a_1 的全部约数即可，剩下的就是 $k = 0$ 的答案。
- 时间复杂度 $O(\sqrt{a})$。

100 分做法

- 当 $k \geq 1$ 时，只有这些数的约数才会对答案产生贡献。

100 分做法

- 当 $k \geq 1$ 时，只有这些数的约数才会对答案产生贡献。
- 求出 m 个数的所有不超过 n 的约数，去重后统计即可。

100 分做法

- 当 $k \geq 1$ 时，只有这些数的约数才会对答案产生贡献。
- 求出 m 个数的所有不超过 n 的约数，去重后统计即可。
- 求出 $k = 1$ 到 m 的所有答案后，剩下的数字个数就是 $k = 0$ 的答案。

100 分做法

- 当 $k \geq 1$ 时，只有这些数的约数才会对答案产生贡献。
- 求出 m 个数的所有不超过 n 的约数，去重后统计即可。
- 求出 $k = 1$ 到 m 的所有答案后，剩下的数字个数就是 $k = 0$ 的答案。
- 时间复杂度 $O(m^2\sqrt{a})$。

Market(market.c/cpp/pas)

给定 n 个物品，每个物品要么不选，要么选一个，第 i 个物品的价格为 c_i，价值为 v_i，出现时间为 t_i。

Market(market.c/cpp/pas)

给定 n 个物品，每个物品要么不选，要么选一个，第 i 个物品的价格为 c_i，价值为 v_i，出现时间为 t_i。

m 次询问，每次询问在出现时间不超过 T_i 的所有物品中选若干件，总花费不超过 M_i 的情况下，被选择物品的价值和的最大值是多少。

Market(market.c/cpp/pas)

给定 n 个物品，每个物品要么不选，要么选一个，第 i 个物品的价格为 c_i，价值为 v_i，出现时间为 t_i。

m 次询问，每次询问在出现时间不超过 T_i 的所有物品中选若干件，总花费不超过 M_i 的情况下，被选择物品的价值和的最大值是多少。

- 测试点 $1, 2$，$n \le 20, m \le 10, c_i, M_i, v_i \le 100$。

Market(market.c/cpp/pas)

给定 n 个物品，每个物品要么不选，要么选一个，第 i 个物品的价格为 c_i，价值为 v_i，出现时间为 t_i。

m 次询问，每次询问在出现时间不超过 T_i 的所有物品中选若干件，总花费不超过 M_i 的情况下，被选择物品的价值和的最大值是多少。

- 测试点 $1, 2$，$n \leq 20, m \leq 10, c_i, M_i, v_i \leq 100$。
- 测试点 $3, 4$，$n \leq 200, m = 1, c_i, M_i, v_i \leq 200, t_i = T_i = 1$。

Market(market.c/cpp/pas)

给定 n 个物品，每个物品要么不选，要么选一个，第 i 个物品的价格为 c_i，价值为 v_i，出现时间为 t_i。

m 次询问，每次询问在出现时间不超过 T_i 的所有物品中选若干件，总花费不超过 M_i 的情况下，被选择物品的价值和的最大值是多少。

- 测试点 $1, 2$，$n \leq 20, m \leq 10, c_i, M_i, v_i \leq 100$。
- 测试点 $3, 4$，$n \leq 200, m = 1, c_i, M_i, v_i \leq 200, t_i = T_i = 1$。
- 测试点 $5, 6$，$n \leq 300, m \leq 100000, c_i, M_i, v_i \leq 300$。

Market(market.c/cpp/pas)

给定 n 个物品，每个物品要么不选，要么选一个，第 i 个物品的价格为 c_i，价值为 v_i，出现时间为 t_i。

m 次询问，每次询问在出现时间不超过 T_i 的所有物品中选若干件，总花费不超过 M_i 的情况下，被选择物品的价值和的最大值是多少。

- 测试点 $1, 2$，$n \leq 20, m \leq 10, c_i, M_i, v_i \leq 100$。
- 测试点 $3, 4$，$n \leq 200, m = 1, c_i, M_i, v_i \leq 200, t_i = T_i = 1$。
- 测试点 $5, 6$，$n \leq 300, m \leq 100000, c_i, M_i, v_i \leq 300$。
- 测试点 7，$n \leq 20, m \leq 100000, c_i, M_i \leq 10^9, v_i \leq 300$。

Market(market.c/cpp/pas)

给定 n 个物品，每个物品要么不选，要么选一个，第 i 个物品的价格为 c_i，价值为 v_i，出现时间为 t_i。

m 次询问，每次询问在出现时间不超过 T_i 的所有物品中选若干件，总花费不超过 M_i 的情况下，被选择物品的价值和的最大值是多少。

- 测试点 $1, 2$，$n \leq 20, m \leq 10, c_i, M_i, v_i \leq 100$。
- 测试点 $3, 4$，$n \leq 200, m = 1, c_i, M_i, v_i \leq 200, t_i = T_i = 1$。
- 测试点 $5, 6$，$n \leq 300, m \leq 100000, c_i, M_i, v_i \leq 300$。
- 测试点 7，$n \leq 20, m \leq 100000, c_i, M_i \leq 10^9, v_i \leq 300$。
- 测试点 $8, 9, 10$，$n \leq 300, m \leq 100000, c_i, M_i \leq 10^9, v_i \leq 300$。

20 分做法 1

- 测试点 $1, 2$ ，$n \leq 20, m \leq 10, c_i, M_i, v_i \leq 100$。

20 分做法 1

- 测试点 $1,2$，$n \le 20, m \le 10, c_i, M_i, v_i \le 100$。
- 对于每个询问，暴力枚举所有选择情况即可。

20 分做法 1

- 测试点 $1, 2$ ，$n \leq 20, m \leq 10, c_i, M_i, v_i \leq 100$。
- 对于每个询问，暴力枚举所有选择情况即可。
- 时间复杂度 $O(m2^n)$。

20 分做法 2

- 测试点 $3, 4$ ，$n \leq 200, m = 1, c_i, M_i, v_i \leq 200, t_i = T_i = 1$。

20 分做法 2

- 测试点 $3, 4$，$n \le 200, m = 1, c_i, M_i, v_i \le 200, t_i = T_i = 1$。
- 经典 01 背包模型，设 f_i 表示装了 i 块钱的最大价值，DP 即可。

20 分做法 2

- 测试点 $3, 4$，$n \le 200, m = 1, c_i, M_i, v_i \le 200, t_i = T_i = 1$。
- 经典 01 背包模型，设 f_i 表示装了 i 块钱的最大价值，DP 即可。
- 时间复杂度 $O(nM + m)$。

60 分做法

- 测试点 $5, 6$ ，$n \leq 300, m \leq 100000, c_i, M_i, v_i \leq 300$。

60 分做法

- 测试点 $5, 6$，$n \leq 300, m \leq 100000, c_i, M_i, v_i \leq 300$。
- 考虑离线，将物品按出现时间排序，同时将询问按 T 排序。

60 分做法

- 测试点 $5, 6$ ，$n \leq 300, m \leq 100000, c_i, M_i, v_i \leq 300$。
- 考虑离线，将物品按出现时间排序，同时将询问按 T 排序。
- 按 T 从小到大依次回答每个询问，维护一个指针，将允许选择的物品放入背包。

60 分做法

- 测试点 $5, 6$，$n \le 300, m \le 100000, c_i, M_i, v_i \le 300$。
- 考虑离线，将物品按出现时间排序，同时将询问按 T 排序。
- 按 T 从小到大依次回答每个询问，维护一个指针，将允许选择的物品放入背包。
- 时间复杂度 $O(nM + m \log m)$。

10 分做法

- 测试点 7 ，$n \leq 20, m \leq 100000, c_i, M_i \leq 10^9, v_i \leq 300$。

10 分做法

- 测试点 7，$n \leq 20, m \leq 100000, c_i, M_i \leq 10^9, v_i \leq 300$。
- 暴力枚举每个物品的选择情况，将其看成二维平面上坐标为（最晚出现时间，总花费）的点，权值为价值和。

10 分做法

- 测试点 7，$n \le 20, m \le 100000, c_i, M_i \le 10^9, v_i \le 300$。
- 暴力枚举每个物品的选择情况，将其看成二维平面上坐标为（最晚出现时间，总花费）的点，权值为价值和。
- 那么对于每个询问，等价于询问某个点左下角范围内的最大权值，将纵坐标离散化后用二维前缀最值预处理即可。

10 分做法

- 测试点 7，$n \le 20, m \le 100000, c_i, M_i \le 10^9, v_i \le 300$。
- 暴力枚举每个物品的选择情况，将其看成二维平面上坐标为（最晚出现时间，总花费）的点，权值为价值和。
- 那么对于每个询问，等价于询问某个点左下角范围内的最大权值，将纵坐标离散化后用二维前缀最值预处理即可。
- 时间复杂度 $O(n2^n + m\log m)$。

100 分做法

- 依旧考虑离线询问，即可去掉时间的限制。

100 分做法

- 依旧考虑离线询问，即可去掉时间的限制。
- 对每个询问二分答案，需要判定用容量为 M 的背包是否可以装下 mid 的价值。

100 分做法

- 依旧考虑离线询问，即可去掉时间的限制。
- 对每个询问二分答案，需要判定用容量为 M 的背包是否可以装下 mid 的价值。
- 设 f_i 表示装了 i 价值所需的最小容量，g_i 表示 $\min(f_i, f_{i+1}, f_{i+2}, \ldots)$。

100 分做法

- 依旧考虑离线询问，即可去掉时间的限制。
- 对每个询问二分答案，需要判定用容量为 M 的背包是否可以装下 mid 的价值。
- 设 f_i 表示装了 i 价值所需的最小容量，g_i 表示 $\min(f_i, f_{i+1}, f_{i+2}, \ldots)$。
- 那么只需要检查 g_{mid} 是否不超过 M 即可。

100 分做法

- 依旧考虑离线询问，即可去掉时间的限制。
- 对每个询问二分答案，需要判定用容量为 M 的背包是否可以装下 mid 的价值。
- 设 f_i 表示装了 i 价值所需的最小容量，g_i 表示 $\min(f_i, f_{i+1}, f_{i+2}, \ldots)$。
- 那么只需要检查 g_{mid} 是否不超过 M 即可。
- 时间复杂度 $O(n^2 v + m \log m)$。

Dash Speed(speed.c/cpp/pas)

给定一棵有 n 个点的树，每条边有承受区间 $[l_i, r_i]$。

Dash Speed(speed.c/cpp/pas)

给定一棵有 n 个点的树，每条边有承受区间 $[l_i, r_i]$。

m 次询问，每次给定一个值 $speed$，求一条最长的链，使得上面所有边的承受区间都包括 $speed$。

Dash Speed(speed.c/cpp/pas)

给定一棵有 n 个点的树，每条边有承受区间 $[l_i, r_i]$。

m 次询问，每次给定一个值 $speed$，求一条最长的链，使得上面所有边的承受区间都包括 $speed$。

- 对于 20% 的数据，$n, m \le 20$。

Dash Speed(speed.c/cpp/pas)

给定一棵有 n 个点的树，每条边有承受区间 $[l_i, r_i]$。

m 次询问，每次给定一个值 $speed$，求一条最长的链，使得上面所有边的承受区间都包括 $speed$。

- 对于 20% 的数据，$n, m \le 20$。
- 对于另外 20% 的数据，$n, m \le 70000$，$l_i = 1$，树退化成链。

Dash Speed(speed.c/cpp/pas)

给定一棵有 n 个点的树，每条边有承受区间 $[l_i, r_i]$。

m 次询问，每次给定一个值 $speed$，求一条最长的链，使得上面所有边的承受区间都包括 $speed$。

- 对于 20% 的数据，$n, m \leq 20$。
- 对于另外 20% 的数据，$n, m \leq 70000$，$l_i = 1$，树退化成链。
- 对于另外 20% 的数据，$n, m \leq 70000$，树退化成链。

Dash Speed(speed.c/cpp/pas)

给定一棵有 n 个点的树，每条边有承受区间 $[l_i, r_i]$。

m 次询问，每次给定一个值 $speed$，求一条最长的链，使得上面所有边的承受区间都包括 $speed$。

- 对于 20% 的数据，$n, m \leq 20$。
- 对于另外 20% 的数据，$n, m \leq 70000$，$l_i = 1$，树退化成链。
- 对于另外 20% 的数据，$n, m \leq 70000$，树退化成链。
- 对于另外 20% 的数据，$n, m \leq 70000$，$l_i = 1$。

Dash Speed(speed.c/cpp/pas)

给定一棵有 n 个点的树，每条边有承受区间 $[l_i, r_i]$。

m 次询问，每次给定一个值 $speed$，求一条最长的链，使得上面所有边的承受区间都包括 $speed$。

- 对于 20% 的数据，$n, m \leq 20$。
- 对于另外 20% 的数据，$n, m \leq 70000$，$l_i = 1$，树退化成链。
- 对于另外 20% 的数据，$n, m \leq 70000$，树退化成链。
- 对于另外 20% 的数据，$n, m \leq 70000$，$l_i = 1$。
- 对于 100% 的数据，$n, m \leq 70000$。

20 分做法 1

- 对于 20% 的数据，$n, m \leq 20$。

20 分做法 1

- 对于 20% 的数据，$n, m \leq 20$。
- 对于每个询问，暴力枚举所有路径即可。

20 分做法 1

- 对于 20% 的数据，$n, m \leq 20$。
- 对于每个询问，暴力枚举所有路径即可。
- 时间复杂度 $O(mn^2)$。

20 分做法 2

- 对于另外 20% 的数据，$n, m \le 70000$，$l_i = 1$，树退化成链。

20 分做法 2

- 对于另外 20% 的数据，$n, m \leq 70000$，$l_i = 1$，树退化成链。
- 将所有边按 r 从小到大排序后依次加入，那么当前的最长链就是最长区间的长度。

20 分做法 2

- 对于另外 20% 的数据，$n, m \leq 70000$，$l_i = 1$，树退化成链。
- 将所有边按 r 从小到大排序后依次加入，那么当前的最长链就是最长区间的长度。
- 并查集维护即可。

20 分做法 2

- 对于另外 20% 的数据，$n, m \le 70000$，$l_i = 1$，树退化成链。
- 将所有边按 r 从小到大排序后依次加入，那么当前的最长链就是最长区间的长度。
- 并查集维护即可。
- 时间复杂度 $O(n \log n)$。

40 分做法 1

- 对于另外 20% 的数据，$n, m \leq 70000$，树退化成链。

40 分做法 1

- 对于另外 20% 的数据，$n, m \leq 70000$，树退化成链。
- 考虑将每条边拆成两个事件：

40 分做法 1

- 对于另外 20% 的数据，$n, m \le 70000$，树退化成链。
- 考虑将每条边拆成两个事件：
- 1. 在 l_i 处把当前位置染黑。

40 分做法 1

- 对于另外 20% 的数据，$n, m \leq 70000$，树退化成链。
- 考虑将每条边拆成两个事件：
- 1. 在 l_i 处把当前位置染黑。
- 2. 在 $r_i + 1$ 处把当前位置染白。

40 分做法 1

- 对于另外 20% 的数据，$n, m \leq 70000$，树退化成链。
- 考虑将每条边拆成两个事件：
- 1. 在 l_i 处把当前位置染黑。
- 2. 在 $r_i + 1$ 处把当前位置染白。
- 然后从左往右依次考虑每个事件，那么当前的最长链就是最长的全黑区间的长度。

40 分做法 1

- 对于另外 20% 的数据，$n, m \le 70000$，树退化成链。
- 考虑将每条边拆成两个事件：
- 1. 在 l_i 处把当前位置染黑。
- 2. 在 $r_i + 1$ 处把当前位置染白。
- 然后从左往右依次考虑每个事件，那么当前的最长链就是最长的全黑区间的长度。
- 用线段树维护最长的全黑区间长度即可，每个节点保存最长全黑区间、最长全黑前缀和最长全黑后缀即可。

40 分做法 1

- 对于另外 20% 的数据，$n, m \le 70000$，树退化成链。
- 考虑将每条边拆成两个事件：
- 1. 在 l_i 处把当前位置染黑。
- 2. 在 $r_i + 1$ 处把当前位置染白。
- 然后从左往右依次考虑每个事件，那么当前的最长链就是最长的全黑区间的长度。
- 用线段树维护最长的全黑区间长度即可，每个节点保存最长全黑区间、最长全黑前缀和最长全黑后缀即可。
- 时间复杂度 $O(n \log n)$。

40 分做法 2

- 对于另外 20% 的数据，$n, m \leq 70000$，$l_i = 1$。

40 分做法 2

- 对于另外 20% 的数据，$n, m \leq 70000$，$l_i = 1$。
- 将所有边按 r 从小到大排序后依次加入，需要维护这个森林的最长链。

40 分做法 2

- 对于另外 20% 的数据，$n, m \le 70000$，$l_i = 1$。
- 将所有边按 r 从小到大排序后依次加入，需要维护这个森林的最长链。
- 对于每个连通块维护其最长链，那么合并两个连通块的时候，新的最长链一定来自于那 4 个点，分 6 种情况取最优解即可。

40 分做法 2

- 对于另外 20% 的数据，$n, m \le 70000$，$l_i = 1$。
- 将所有边按 r 从小到大排序后依次加入，需要维护这个森林的最长链。
- 对于每个连通块维护其最长链，那么合并两个连通块的时候，新的最长链一定来自于那 4 个点，分 6 种情况取最优解即可。
- 并查集维护。

40 分做法 2

- 对于另外 20% 的数据，$n, m \leq 70000$，$l_i = 1$。
- 将所有边按 r 从小到大排序后依次加入，需要维护这个森林的最长链。
- 对于每个连通块维护其最长链，那么合并两个连通块的时候，新的最长链一定来自于那 4 个点，分 6 种情况取最优解即可。
- 并查集维护。
- 时间复杂度 $O(n \log n)$。

100 分做法

- 如果像序列上一样扫描的话，带加边、删边的动态树上最长链并不是很好做。

100 分做法

- 如果像序列上一样扫描的话，带加边、删边的动态树上最长链并不是很好做。
- 考虑分治，假设当前分治区间为 $[L, R]$，那么对于所有承受区间完全包含 $[L, R]$ 的边，都可以在这个时候加入，剩下的边则按照它与 $mid = \lfloor \frac{L+R}{2} \rfloor$ 的关系划分到下一层递归分治。

100 分做法

- 如果像序列上一样扫描的话，带加边、删边的动态树上最长链并不是很好做。
- 考虑分治，假设当前分治区间为 $[L, R]$，那么对于所有承受区间完全包含 $[L, R]$ 的边，都可以在这个时候加入，剩下的边则按照它与 $mid = \lfloor \frac{L+R}{2} \rfloor$ 的关系划分到下一层递归分治。
- 因为分治完一侧后需要分治另一侧，所以这里的并查集不能路径压缩，而应该按秩合并，同时按时间顺序记录下所有修改操作，在回溯的时候撤销即可。

100 分做法

- 如果像序列上一样扫描的话，带加边、删边的动态树上最长链并不是很好做。
- 考虑分治，假设当前分治区间为 $[L, R]$，那么对于所有承受区间完全包含 $[L, R]$ 的边，都可以在这个时候加入，剩下的边则按照它与 $mid = \lfloor \frac{L+R}{2} \rfloor$ 的关系划分到下一层递归分治。
- 因为分治完一侧后需要分治另一侧，所以这里的并查集不能路径压缩，而应该按秩合并，同时按时间顺序记录下所有修改操作，在回溯的时候撤销即可。
- 时间复杂度 $O(n \log^2 n)$。

Thank you!