

AtCoder Beginner Contest 137

2020 年 5 月 3 日

A: +-x

考察

問題文の通りに $A + B$, $A - B$, $A \times B$ の中で最大の数を出力する。

実装

`std::max` は $\max(\{A, B, C\})$ のように指定することで 2 個以上の引数も受け付けることができる。

B: One Clue

考察

黒で塗られている石が置かれる可能性のある座標 i は $X - K + 1 \leq i \leq X + K - 1$ であり i を出力する。

C: Green Bin

考察

文字列全てをそれぞれソートすると、アナグラム関係にある文字列はイコールの関係になる。イコールの関係になる文字列を数え上げ、組み合わせの総和が解答となる。

実装

文字列 s を全て配列に格納していたが、この問題の場合アナグラムの種類ごとの数さえ管理されていれば良いので、個々の s を標準入力で受け付けた後に、処理してしまえばよい。

Summer Vacation

考察

まず考えたのは、報酬の最大値は i 日目の最大値は、 $i - 1$ 日目の最大値に選択可能な仕事で最大の報酬を加えたものになるので、動的計画法で解答可能と考えた。ただ、 $O(NM)$ となるので間に合わない。

そこで、貪欲法 ベースで考える方針とした。前方から考えると報酬と報酬受け取り期間の 2 つの自由度となってしまうので選ぶのが難しくなる。一方で、後方から考えると報酬のみ考えればよくなり選びやすいため後方から考えることにする。なぜそれでうまくいくのか証明はできていないため不安要素は残るが、反例が思いつかないのでよしとする。

実装

現在日で報酬を受け取れる仕事の中から最大の報酬を選ぶようにする。そこで、`std::priority_queue` で報酬を管理するようにする。 N 件の仕事を `queue` に格納し、 M 回取り出すため、計算量は $O(M + N \log N)$ となり間に合う。