ABC 383 B - Humidifier 2

hiragn

2024 年 12 月 14 日

1. 問題の概要

> #と. からなる H 行 W 列のマス目でオフィスをあらわす。#は机で. は床である。
> 異なる 2 つの床のマスを選んで加湿器を置く。加湿器からのマンハッタン距離が D 以下のマスは加湿される。
> 加湿される床のマスの個数の最大値を求めよ。
>
> https://atcoder.jp/contests/abc383/tasks/abc383_b

2. 解法

普通のプログラミング言語だと 2 台の加湿器と床の 3 点を考える必要があり，6 重のループが必要になる。これを Select で処理したい。

部屋の様子は文字列のリストして与えられる。まず Characters と Position を使って，床の座標を取り出す。

```
In[]:= room1 = {".###.", ".#.##"}
Out[]= {".###.", ".#.##"}

In[]:= room2 = Characters /@ room1
Out[]= {{".", "#", "#", "#", "."}, {".", "#", ".", "#", "#"}}

In[]:= Position[room2, "."]
Out[]= {{1, 1}, {1, 5}, {2, 1}, {2, 3}}
```

マンハッタン距離は組み込み関数の ManhattanDistance で処理できる。

床座標のリストから Subsets で 2 つの点を取り出して，それらからのマンハッタン距離が D 以下の点を Select して個数を数えればよさそう。サンプル 1 でやってみた。

```
In[]:= Clear["Global`*"];
h = 2; w = 5; d = 1;
room = {".###.", ".#.##"};
room = Position[Characters /@ room, "."];
cond[u_, v_] := ManhattanDistance[u, v] <= d;
res = 0;
Do[{u, v} = x;
  cnt = Length@Select[room, cond[u, #] || cond[v, #] &];
  res = Max[cnt, res], {x, Subsets[room, {2}]}];
res

Out[]= 3
```

このままでもいいが関数型言語らしい，ループしない形に書き直した。

```
In[]:= Clear["Global`*"];
h = 2; w = 5; d = 1;
room = {".###.", ".#.##"};
room = Position[Characters /@ room, "."];
cond[u_, v_] := ManhattanDistance[u, v] <= d;
calc[{u_, v_}] := Length@Select[room, cond[u, #] || cond[v, #] &];
ans = Max[calc /@ Subsets[room, {2}]]

Out[]= 3
```

これを 1 つの関数にまとめて完成。サンプルもぜんぶ通った。

```
In[]:= Clear["Global`*"];
solve[{w_, h_, d_, room_}] := Module[{pos},
   pos = Position[Characters /@ room, "."];
   cond[u_, v_] := ManhattanDistance[u, v] <= d;
   calc[{u_, v_}] := Length@Select[pos, cond[u, #] || cond[v, #] &];
   ans = Max[calc /@ Subsets[pos, {2}]]];

case1 = {2, 5, 1, {".###.", ".#.##"}};
case2 = {5, 5, 2, {".#.#.", ".....", ".#.#.", "#.#.#", "....."}};
case3 = {4, 4, 2, {"....", ".##.", ".##.", "...."}};
solve /@ {case1, case2, case3} == {3, 15, 10}

Out[]= True
```