Проблем 4. Менхетн

Међу датим скупом тачака $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ исписати најмање менхетн растојање међу њима.

Менхетн растојање између тачака (a, b) и (c, d) се дефинише као $|a - c| + |b - d|$.

Улаз. (Улазни подаци се учитавају на стандардног улаза) У првом реду улаза налази се природни број n ($2 \leq n \leq 1000$). У наредних n редова се учитавају координате тачака: у $(k+1)$-ов реду се учитава тачка (x_k, y_k) ($-100.000 \leq x_k, y_k \leq 100.000$), где су x_k и y_k цели бројеви.

Излаз. (Излазне подаци се исписују на стандардни излаз) У првом и јединoм реду исписати најмање растојање.

Пример 1.

menhetn.in	menhetn.out
5 1 2 4 2 8 3 9 5 15 11	3

Објашњење. Међу свим паровима тачака на најмањем растојању се налазе прва и друга ($|1-4| + |2-2| = 3$), и трећа и четврта тачка ($|8-9| + |3-5| = 3$).

Пример 2.

menhetn.in	menhetn.out
2 -1 1 -1 1	0

Ограничења. Временско ограничење: $0.1s$; Меморијско ограничење: $16MB$

Решење и анализа. Проблем Менхетн се показао као најлакши проблем на овим квалификацијама, где под најлакшим подразумевамо просечан број освојених бодова. Наиме, просечан број бодова на овом проблему је био 53 док је око 60% такмичара имало позитиван скор на њему.

Како је ограничење броја тачака 1.000, испитивање растојања сваког пара тачака посебно ће задовољити дато временско ограничења. Заиста, број могућих парова једнак је $\binom{n}{2}$ што је реда величине n^2. Како је само рачунање растојања за дате две тачке константне сложености (само једна операција), сложеност овог алгоритма је $O(n^2)$.

Означимо дати низ тачака са a. За имплементацију ове идеје потребна су нам два показивача i и j. Они ће показивати на тачке у низу a (горе описани пар тачака) које тренутно испитујемо. Како је све једно да ли посматрамо тачке $a[i]$ и $a[j]$ или $a[j]$ и $a[i]$, можемо (без губљења општости) узети да је $i < j$. Једино на шта треба обратити пажњу јесте да индекси i и j морају бити различити (иначе би увек добили растојање 0 као решење). Због горње претпоставке да је $i < j$, индекс i треба "прошетати" од 1 до n а за j узимати вредности од $i + 1$ до n. Ово се импелемтира угњежденим петљама по i и j.

У сваком кораку рачунамо растојање између тачака $a[i]$ и $a[j]$ (у псеудо коду означено са *currentDist*). Уколико је вредност овог растојања мања од вредности тренуног најмањег растојања (*minDist* у псеудо коду) најмање растојања постављамо управо на ову вредност.

Алгоритам: Псеудо код проблема Менхетн

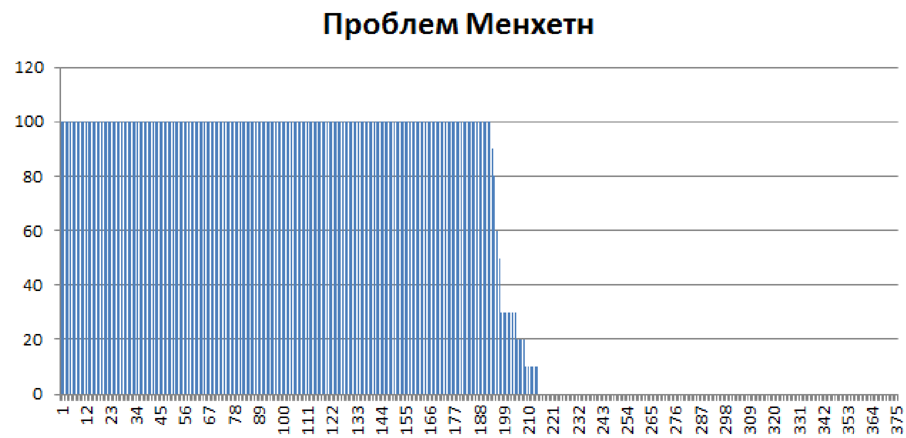
```
Input: низ тачака a дужине n
Output: најмање Менхетн растојања међу датим тачкама

minDist = ∞;
for i ← 1 to n do
    for j ← i + 1 to n do
        currentDist = |a[i].x − a[j].x| + |a[i].y − a[j].y|;
        if (currentDist < minDist) then
            minDist = currentDist;
        end
    end
end

return minDist
```

Напоменимо и то да је решење које испитује све парове $a[i]$ и $a[j]$, за које је $i \neq j$, коректно. Свако растојање би рачунали два пута, због симетрије, али то не би утицало на сложеност алгоритма. Међутим, уколико је неко убрзање лако увести (као у овом случају где је чак овај убрзнани приступ једноставнији) свакако неће шкодити да исти имплементирате. У неким напреднијим проблемима ово може много значити.

У псеудо коду је иницијална вредност промењиве *minDist* постављена на ∞. Наравно, ово представља само нотацију у псеудо коду. Под бесконачношћу се подразумева било који број који је сигурно већи од свих могућих растојања међу овим тачкама. Како је ограничења за координате једнако 100.000 по апсолутној вредности, имамо да је максимално растојање које можемо очекивати једнако 400.000. Ово растојање се добија као растојање између тачака $(-x, -x)$ и (x, x) где је $x = 100.000$. Конкретан случај, као "најгори могући случај", је за дати проблем било једноставно наћи. То наравно није правило. Зато се увек треба оградити од евентуалне грешке постављањем ове вредности на већу (то свакако неће погоршати ствари).

Напомена. Интересантно је и напоменути зашто се овај проблем зове баш Менхетн (наравно ово није случајност). Наиме, овде смо рачунали растојања тачака у равни као апсолутно растојање по координатана (на први поглед неком делује чудно). Свима је вероватно познато стандардно растојање дефинисано као $\sqrt{(a.x - b.x)^2 + (a.y - b.y)^2}$ - познатије као Еуклидско растојање. Наиме, Еуклидско растојање представља најкраћи пут између две тачке у равни (баш зато што је праволинијски). Међутим, постоје и друге метрике (ово је формални израз за растојање над скупом тачака), а Менхетн је једна од њих. Други назив за ову метрику је Такси метрика (енг. *taxycab metric*) односно растојање. Назив потиче из начина кретања таксиста по блоковима града Менхетна који чине решетку (матрицу). Другим речима и ова метрика представља најкраће растојање између тачака али у "решетки".

Грешке такмичара. На крају, направимо кратак осврт на неке од грешака на које смо наишли. Најфреквентнија грешка била је везана за тип промењиве. Наиме, многи такмичари који су куцали у *Pascal*-у су за тип узимали *Integer* уместо *LongInt*. Међутим, овај тип узима вредности из сегмента $[-2^{15}, 2^{15} - 1]$ који не обухвата ограничења координата дата у тексту проблема. Тако су многи, уместо максималних 100, освојили једва 30 бодова на овом задатку.



Слика 1. График броја освојених бодова свих такмичара.

Мало мање учесталија грешка била је везана управо за иницијалну вредност минималног растојања тј. бесконачност о којој смо мало пре дискутовали. Многи су, не удубљујући се много у разматрање могућих вредности растојања, постављали $\infty = 200.000$ (вероватно посматрајући само позитивне координате). Горе наведена напомена, да се ова вредност увек "мало" повећа како би се избегли оваки и неки други специјални гранични случајеви, би решила овај проблем.