

Informatikából kitűzött feladatok

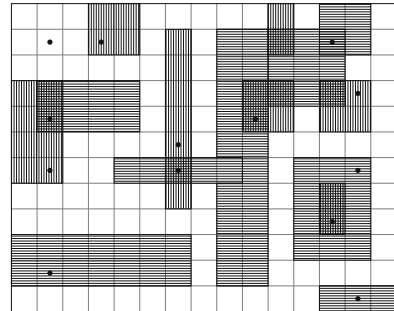


I. 460. Egy téglalap alakú üveglapra gondolatban egy $N \times M$ -es ($10 \leq N, M \leq 10\,000$) négyzethálót helyezünk. Az üveglapot rá merőlegesen P ($0 \leq P \leq N \times M$) pontban egy-egy nem polarizált fénysugárral megvilágítjuk úgy, hogy minden fénysugár pontosan egy négyzetre essen. Az üveglapra K ($0 \leq K \leq 1\,000$) darab különböző szélességű és hosszúságú, téglalap alakú polárszűrőt helyezünk el úgy, hogy oldalaik a négyzetháló rácsvonalaira esnek. A polárszűrők érintkezhetnek és átfedhetnek egymást, de a lapról nem lóghatnak le. Kétféle polárszűrő van: az egyik a beérkező fényt az üveglap felső oldalával párhuzamosan, a másik arra merőlegesen polarizálja. Az a fénysugár, amely két, egymásra merőlegesen polarizáló polárszűrőre esik, nem jut át az üveglapon.

Készítsünk programot **i460** néven, amely a következő problémákat oldja meg.

A program olvassa be a standard input első sorából N -et, M -et, P -t és K -t. A következő P sorból a megvilágított négyzetek bal alsó sarkának koordinátáit, utána K sorból a polárszűrők bal alsó, illetve jobb felső sarkának koordinátáit és a polarizációt (p vagy m). A koordináták egész számok, az üveglap bal alsó sarkának koordinátái $1, 1$. A program írja a standard output egymás utáni három sorába a következő feladatok megoldását:

- soroljuk fel a beolvasás sorrendjében azoknak a fényforrásoknak a sorszámát, amelyek fénye nem jut át az üveglapon a polárszűrők miatt;
- adjuk meg, hány olyan négyzet van, amelyet nem világítunk meg, de a polárszűrők miatt nem átlátszó;
- adjuk meg, melyek azok a polárszűrők, amelyeket eltávolítva a fénysugarak átlépése nem változik (több lehetséges megoldás esetén elég egyet megadni).



Példa:

Standard bemenet (a / jel újsor karaktert jelöl):	Standard kimenet:
15 12 13 15	3 6 7 8
2 2 / 2 6 / 2 8 / 2 11 / 4 11 / 7 6 / 7 7	6
10 8 / 13 4 / 13 11 / 14 1 / 14 6 / 14 9	1 4 8 12
1 2 8 4 p / 1 6 3 10 m / 2 8 6 10 p / 4 11 6 13 m	
5 6 10 7 p / 7 5 8 12 m / 9 2 11 11 p / 9 9 14 12 p	
10 8 12 10 m / 11 11 12 13 m / 12 3 15 7 p / 13 1 16 2 p	
13 4 14 6 m / 13 8 15 10 m / 13 11 15 13 p	

Beküldendő egy tömörített `i460.zip` állományban a program forráskódja és rövid dokumentációja, amely megadja, hogy a forrásállomány melyik fejlesztői környezetben fordítható.

I. 461 (É). Hanna és Panna palacsintázót üzemeltet. Szeretnének mindenkit friss palacsintával kiszolgálni, ezért a hozzávalókat egész napra előre elkészítik, és a palacsintát magát akkor sütik, ha valaki éppen kér. A vevő belép az üzletbe, rendel, és rendelését – a többiek rendelésének teljesítése után – azonnal elkészítik.

Feladatunk, hogy ezt a folyamatot táblázatkezelővel modellezzük. Ehhez egy munkafüzet négy munkalapját használjuk. A **Vásárlás** munkalap tartalmazza a vevő érkezési idejét, a rendelés adatait, a fizetett összeget és a kiszolgálás időpontját. A **Palacsinták** munkalapon a palacsinták ízesítése, ára és az a darabszám szerepel, amennyihez elegendő töltelék áll rendelkezésre. Az **Üzleti adatok** munkalap tartalmazza a nyitás és zárás időpontját, valamint a **Vásárlás** munkalap kapcsán lentebb említett T , Db , t értékeket. A **Segéd** munkalapon a feladat megoldásához szükséges segédszámításokat végezhetjük.

A **Vásárlás** munkalap kitöltését az alábbiak szerint készítjük, legfeljebb 1000 vásárlóra számítva:

1. Az **érkezés** oszlopban a vevők belépési időpontja szerepel. Az időpontokat véletlenszerűen állítsuk elő. A vevők nyitástól zárásig legfeljebb másodpercnyi sűrűséggel követhetik egymást, de feltételezhetjük, hogy T percnél tovább nem marad új vásárló nélkül az üzlet.
2. A **darabszám** oszlop a vevő által rendelt mennyiséget tartalmazza. Egy vevő csak egyfajta palacsintát kérhet, abból is legfeljebb Db darabot. Értékét véletlenszerűen kell előállítanunk.
3. Az **ízesítés** oszlopban jelenjen meg, hogy milyen ízesítésűt választott. Értékét véletlenszerűen állítsuk elő úgy, hogy a **Palacsinta** munkalapon szereplő minden ízesítésnek azonos valószínűsége legyen.
4. A **fizetés** oszlopban a rendelt darabszámtól és ízesítéstől függő értéket kell megjelenítenünk. Figyelembe kell vennünk azonban, hogy ha az adott ízesítésű palacsintához már nem áll rendelkezésre elegendő töltelék, akkor csak az elkészített palacsinták után fizet a vevő.
5. A **kiszolgálás** oszlopban a kiszolgálás időpontját kell meghatároznunk az alábbiak ismeretében. A palacsintákat egyesével, egymás után sütik. Egy palacsinta elkészítéséhez szükséges idő t másodperc. Sütése akkor kezdődik el, ha minden korábbi vevő kiszolgálása megtörtént. A betérő vásárló biztosan kivárja a sorát, valamint ha zárás előtt lépett az üzletbe, biztosan kiszolgálják. A rendelés, fizetés időigényétől eltekintünk.
6. A **Vásárlás** munkalapon csak a zárás előtt érkező vásárlók sorainak adatai jelenjenek meg. Annak a vásárlónak a sora, aki utoljára kapott a választott ízesítésből, sárga, akit pedig egyáltalán nem tudtak kiszolgálni, szürke háttérrel látszódjon.

A számítások során egész másodpercekkel dolgozunk, a számot tartalmazó cellákban egész értékek szerepeljenek.

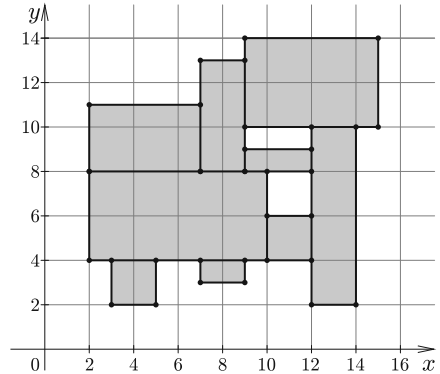
Beküldendő egy `i461.zip` tömörített állományban a megoldást tartalmazó `i461` munkafüzet és a dokumentációt magában foglaló `i461.pdf` fájl. A dokumentáció tartalmazza a használt táblázatkezelő nevét és verziószámát.

I. 462. Az irodai munkában gyakran előfordul, hogy ugyanazt a részfeladatot számtalanszor kell végrehajtani. A „rabszolgamunka” automatizálásával sok időt lehet megspórolni. Egy ilyen gyakran ismétlődő probléma, hogy egy munkafüzet lapjain található adatok alapján kell diagramot készíteni.

Jelen feladatban legfeljebb 100 munkalappal kell dolgoznunk. Minden munkalap első sora tartalmazza a kategóriák nevét, az első oszlop pedig az adatsorok feliratát. A munkalapok legfeljebb 12 sornyi és 12 oszloponyi adatot tartalmaznak. Az összes munkalapra kell készítenünk egy halmazott oszlopdigrammot az ott található adatokból. A diagram címe az A1-es cella tartalma legyen. Az összes diagramot egyetlen makróindítással kell elkészítenünk.

Beküldendő egy `i462.zip` tömörített állományban a megoldást tartalmazó `i462` munkafüzet és a dokumentációt magában foglaló `i462.pdf` fájl. A dokumentáció tartalmazza a használt táblázatkezelő nevét, verziószámát és a megoldást jelentő makró(k) lényeges elemeinek magyarázatát és indításának módját.

I/S. 28. Egy hatalmas telekre sportkomplexumot terveznek. A telekre gondolatban egy koordináta-rendszert helyeznek. A kialakítandó N darab sportpálya mind téglalap alakú, oldalaik a koordináta-rendszer tengelyeivel párhuzamosak. Ismerjük minden pálya két szemközti csúcsának koordinátáit. A pályáknak nincs közös területe, de oldalaik érintkezhetnek egymással. A tervek szerint két pálya között pontosan akkor készül majd átjáró, ha legalább D hosszú szakaszon érintkeznek egymással. Egy pálya egy oldalát a vele érintkező más pályák több szakaszra osztják (ha nem, akkor a teljes oldalt egy szakasznak tekintjük). Vegyük azokat a szakaszokat, amik a külső térre néznek, vagyis nem részei más pálya oldalának. Egy-egy ilyen, legalább D hosszú szakaszra bejáratot terveznek. Adjuk meg, hogy a sportkomplexum tervében összesen hány bejárat és hány átjáró van.



Bemenet: az első sor a pályák N számát és a D hosszúságot tartalmazza. A következő N sor mindegyike négy egész számot tartalmaz: egy-egy pálya két szemközti csúcsának koordinátáit.

Kimenet: egy sorba írjuk ki az átjárók, aztán a bejáratok számát.

Bemenet (a / jel sortörést helyettesít)	Kimenet
9 2	9 22
2 8 10 4 / 7 13 9 8 / 9 9 12 8 / 10 6 12 4 / 12 10 14 2	
9 14 15 10 / 2 11 7 8 / 3 4 5 2 / 7 4 9 3	

Korlátok: $1 \leq N \leq 10^5$, $-10^9 \leq$ koordináták $\leq 10^9$, $1 \leq D \leq 10^9$, egész számok.

Értékelés: a pontok 20%-a kapható, ha $D = 1$; további 20% kapható, ha a pályák 1×1 -es négyzetek; további 20% kapható, ha $N \leq 1000$; további 40% kapható az eredeti korlátokra.

Időlimit: 0,3 mp, memórialimit: 100 MiB.

S. 127. Egy országban N darab város van, amiket kétirányú utak kötnek össze. Az úthálózatra teljesül, hogy bármelyik városból bármelyikbe el lehet jutni. Két város között legfeljebb egy közvetlen út van. Minden útra adott egy súlykorlátozás, hogy rakománnyal együtt legfeljebb mekkora tömegű teherautó mehet rajta. Egy áruszállító cég nyersanyagokat szállít városok között. A cégnek nem számít, hogy két város között a szállítás milyen hosszú úton történik, de a lehető legtöbb nyersanyagot szeretnék elvinni egy-egy teherautóval. Adott Q darab várospár, amik között nyersanyagot kell szállítani. Minden várospárra adjuk meg, hogy legfeljebb milyen nehéz lehet a teherautó szállítmánnyal.

Bemenet: az első sor tartalmazza a városok N számát, az utak M számát és a várospárok Q számát. A városokat 0-tól $(N - 1)$ -ig indexeljük. A következő M sor mindegyike három számot tartalmaz: egy adott út mely városokat köt össze, és mekkora rá a súlykorlátozás. A következő Q sor mindegyike két számot tartalmaz: egy adott várospár indexeit, amik között szállítani kell.

Kimenet: Q sor mindegyikébe egy számot írjunk: azt a maximális súlyt, amilyen nehéz teherautó indulhat az adott várospár között.

Bemenet (a / jel sortörést helyettesít)	Kimenet
6 7 5	4 / 3 / 2 / 5 / 3
5 1 2 / 1 0 3 / 1 4 4 / 2 3 5 / 2 1 3 / 3 4 6 / 0 3 2	
1 4 / 0 3 / 3 5 / 2 4 / 2 0	

Korlátok: $2 \leq N, Q \leq 10^5$, $N - 1 \leq M \leq 5 \cdot 10^5$, $1 \leq$ súlykorlátok $\leq 10^9$, egész számok.

Értékelés: a pontok 20%-a kapható, ha egy útra a súlykorlát csak 1 vagy 2 lehet; további 20% kapható, ha $N \leq 100$; további 20% kapható, ha $M \cdot Q \leq 10^6$; további 40% kapható az eredeti korlátokra.

Időlimit: 0,7 mp, memórialimit: 100 MiB.

A feladatok megoldásai regisztráció után a következő címen tölthetők fel:

<https://www.komal.hu/munkafuzet>

Beküldési határidő: 2018. október 10.