HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

Reiknirit/Algorithms
T- 301 – REIR

Lokapróf / *Final Exam*

I	
I I	
3+4	
V	
VI	
VII	
VIII	
IX	
X	
XI	
XII	
XIII	
Total	

Kennari/*Instructor*: Magnús M. Halldórsson

Dagsetning/*Date*: Þriðjudagur 14. nóvember 2017 /
Tuesday, November 14, 2016

Tími/*Time*: 9.00 – 12.00 / *9:00 – 12:00*

Hjálpargögn/*Supporting materials*:
*** Ein A4 síða skrifuð öðru megin. Handrituð eða 10^{+}pt font /**
Single A4-page (Handwritten or with 10^{+}pt font)
*** Einfaldur vasareiknir / *Simple calculator***
(eða með minni útþurrkað/*or with memory cleared*)

Nafn/*Name*:_____

Kt./*ID*: _____

I. (10%) Tímaflækja / Tilde time complexity

A. Hver er tilda-tímaflækja hverra eftirfarandi kóðabúta, sem fall af N? Veldu fall úr töflunni til hægri. Nota má fall oftar en einu sinni. / *What is the time complexity of the following code segments, as function of N, in tilde notation? Choose a function from the table on the right; a function can be used more than once.*

(a)

```
int x = 1;
for (int j=N; j > 0; j--)
   for (int i=0; i < j; i++)
       x = x*i;
```

Svar/*Answer*: _____

(b)

```
for (int j=N; j >= 0; j--)
   for (int i=N; i > 0; i = i/2)
      count++;
```

Svar/*Answer*: _____

(c)

```
for (int j=N; j > 0; j = j/2)
   for (int i=0; i < j; i++)
      count++;
```

Svar/*Answer*: _____

(d)

```
public int f(int A[])
  { return f(A,0,A.length); }

public int f(int[] A, int L, int H) {
  if (H-L< 1) return 0;
  int sum=0, M=(H+L)/2;
  for (int i=L; i < H; i++)
     sum  += A[i];
  return sum/(H-L) + f(A,L,M) + f(A,M+1,H);
}
```

Svar/*Answer*: _____

(e)

```
public void g(int[] A, int N) {
  if (N < 1) return;
  A[N] = 0; g(A, N-1);
  A[N] = 1; g(A, N-1);
}
```

Svar/*Answer*: _____

log N
N
½ N log N
N log N
2 N log N
½ N^2
N^2
2 N^2
N^3
2^N

II. (8%) Mælingar/Measurements

(a) (6%) Þú mælir keyrslutíma reiknirita, þar sem N er fjöldi staka í inntakinu. / *You measure the elapsed time when running three different algorithms, where N is the number of items in the input.*

Merktu við þann vaxtarhraða sem samsvarar best viðkomandi mælingum. */ Mark the one-word hypothesis on the order of growth of the running time that best matches the experiments.*

(i)

N	1,000	4,000	16,000	64,000
Tími/ *Time*	0.1 s	1.5 s	24 s	1.5 s

línulegt/*linear* ☐ annað veldi/*quadratic* ☐ þriðja veldi/*cubic* ☐ fjórða veldi/*fourth power* ☐

(ii)

N	2,000	3,000	5,000	8,000
Tími/ *Time*	0.9 s	2.0 s	6.0 s	15.9 s

línulegt/*linear* ☐ annað veldi/*quadratic* ☐ þriðja veldi/*cubic* ☐ fjórða veldi/*fourth power* ☐

(iii)

N	1,000	10,000	100,000	1,000,000
Tími/ *Time*	0.0 s	0.4 s	3.8 s	37.4 s

línulegt/*linear* ☐ annað veldi/*quadratic* ☐ þriðja veldi/*cubic* ☐ fjórða veldi/*fourth power* ☐

(b) (2%) Fylki er fyllt af N hlutum af taginu `Wot`. Hve mikið pláss tekur það allt saman? Sýnið með tilda-rithætti, eins einfaldað og mögulegt er. / *An array is filled with N objects of type* `Wot`*. How much memory does it use in total? Express it in tilde-notation, as simplified as possible.*

```
public class Wot {
  private int x,y,z;
}
```

Svar: _____

III. (8%) Hrúgur / *Heaps*

(a) (4%) Hvaða hólf í max-hrúgu með 15 stökum gætu hugsanlega breyst við `insert()` aðgerð? Dragðu hring í kringum þau stök. / *Which entries of a max-heap with 15 elements could possibly be changed during an* `insert()` *operation? Draw a circle around the entries*.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
--	X	x	x	x	x	x	x	x	x	x	x	x	x	x	x	

(b) (4%) Hvaða hólf í max-hrúgu með 16 stökum gætu hugsanlega breyst við `sink(3)` aðgerð? Dragðu hring í kringum þau hólf. / *Which entries of a max-heap with 16 elements could possible change during a* `sink(3)` *operation? Circle the entries.*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
-	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

IV. (4%) Hugtök / Concepts

Dragðu línu frá hverju hugtaki vinstra megin yfir á skilgreiningu hægra megin.
/ *Draw a line from each concept on the left to a definition on the right.*

amortized	tekur pláss þrátt fyrir að vera ekki lengur í notkun / *takes up space in spite of being no longer in use*
	vex eins hratt og $N \log N$ / *order of growth* $N \log N$
reducing	setja fram verkefni út frá lausn á öðru verkefni / *formulating a problem in terms of a solution to another problem*
	þjöppun á gagnagrind / *compression of a data structure*
open addressing	nýta tóm hólf til að glíma við árekstra / *relying on empty entries to help with collision resolution*
	meðaltal yfir runu aðgerða / *average over a number of operations*
linearithmic	varðveita innri röðun jafngildra lykla / *preserving the sorted order of equal keys*

V. (10%) Röðunarreiknirit/ Sorting algorithms

Í töflunni að neðan er dálkurinn lengst til vinstri inntak af strengjum og dálkurinn lengst til hægri strengirnir raðaðir. Hver af hinum níu dálkunum er staðan í miðri keyrslu á einu af röðunarreikniritunum (númer 2-10) listuð til hægri. Parið saman dálkana og reikniritin, og notið hverja tölu einu sinni. / *The leftmost column in the table below is the original input of strings to be sorted, and the rightmost column consists of the strings in sorted order. The other nine columns are the contents at some intermediate step during one of the sorting algorithms (number 2-10) listed on the right. Match each algorithm by writing its number under the corresponding column. Use each number exactly once.*

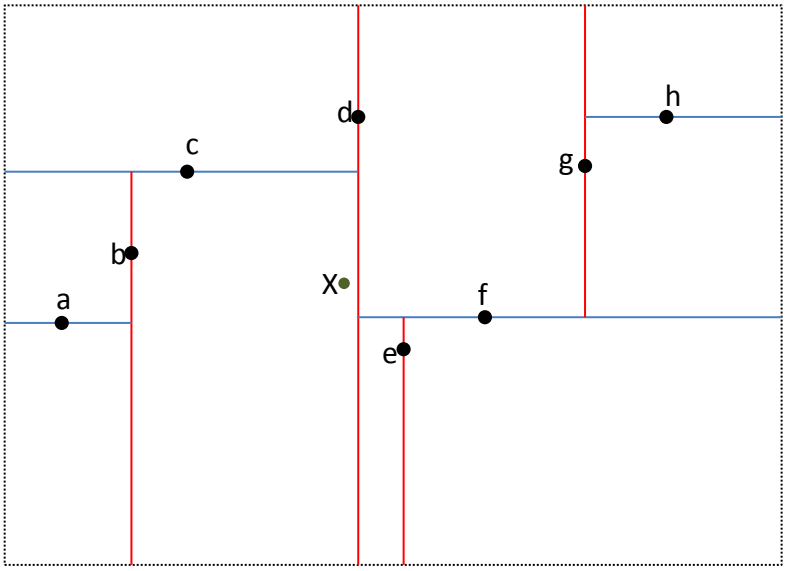
```
skor rass anda anda geim slak anda anda varp geim anda
slak rauk blak blak rass blak blak blak taka rauk blak
rass slak kast brun blak skak brun brun takt rass brun
rauk skor keil drif klif jobs gang drif svig hopp drif
blak anda klif gang anda anda kast gang svif blak gang
klif blak lykt geim keil lyft keil geim skak klif geim
anda keil rass hnit kast klif klif hnit rauk anda hnit
keil klif rauk hopp hopp svif lykt hopp rauk keil hopp
renn kast renn jobs lykt drif rass kast spil hnit kast
kast lykt slak klif gang svig rauk keil slak kast keil
svif renn skor keil brun stik rauk klif stik drif klif
lykt svif svif kast hnit leik rauk leik rass lykt leik
rauk brun rauk lykt drif keil renn rauk rauk leik lyft
rauk gang rauk lyft lyft spil slak rauk hopp lyft lykt
gang rauk gang leik leik geim skor slak rauk gang rass
brun rauk brun rass rauk hnit svif rass rauk brun rauk
svig lyft svig rauk rauk taka leik svig keil rauk rauk
spil spil spil renn rauk lykt lyft spil renn spil rauk
lyft svig lyft rauk rauk takt jobs lyft lyft rauk rauk
varp varp varp rauk rauk gang spil varp kast varp rauk
taka leik taka rauk jobs renn stik taka blak taka jobs
jobs jobs jobs rauk stik skor svig jobs jobs jobs renn
stik stik stik skor taka hopp taka stik skor stik slak
leik taka leik slak skak varp varp lykt leik rauk skak
skak drif skak svif takt rass drif skak lykt skak skor
takt hnit takt svig varp kast geim takt klif takt spil
drif skak drif spil spil rauk hnit rauk drif svif stik
hnit takt hnit stik svig rauk hopp skor hnit renn svif
hopp geim hopp skak svif rauk rauk renn anda rauk svig
rauk hopp rauk taka renn rauk rauk rauk gang slak taka
geim rauk geim takt slak rauk skak svif geim skor takt
rauk rauk rauk varp skor brun takt rauk brun svig varp

 0                                                   1
---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
```

(0) Original input
(1) Sorted
(2) Selection sort
(3) Insertion sort
(4) Mergesort (top-down)
(5) Mergesort (bottom-up)
(6) Quicksort (standard, no shuffle, Median-of-3 pivots)
(7) Quicksort (3-way, no shuffle, Median-of-3 pivots)
(8) Heapsort
(9) LSD radix sort
(10) MSD radix sort

VI. (5%) k-D tré / k-D trees

Punktunum a til h hefur verið stungið inn í 2-d tréið að neðan. (Nöfn þeirra samsvara ekki lyklunum.)
/ The points a through h have been inserted into the 2-d tree below. (Their labels don't correspond to their keys.)



Framkvæmd er `nearest neighbor` aðferðin úr S2-verkefninu á punktinn merktann X. Listaðu nóðurnar sem heimsóttar eru, í þeirri röð sem þær eru heimsóttar.
/ `Nearest neighbor` search is performed on the point marked X, using the method from assignment S2. List the trees nodes visited, in the order that they are visited.

_____

VII. (6%) Rauð-svört tré / *Red-black trees*

A. (6%) Stingdu eftirfarandi lyklum í gefinni röð inn í tómt vinstri-vísandi rautt-svart tré (LLRB).
/ *Insert the following keys in that order into an empty left-leaning red-black tree (LLRB).*

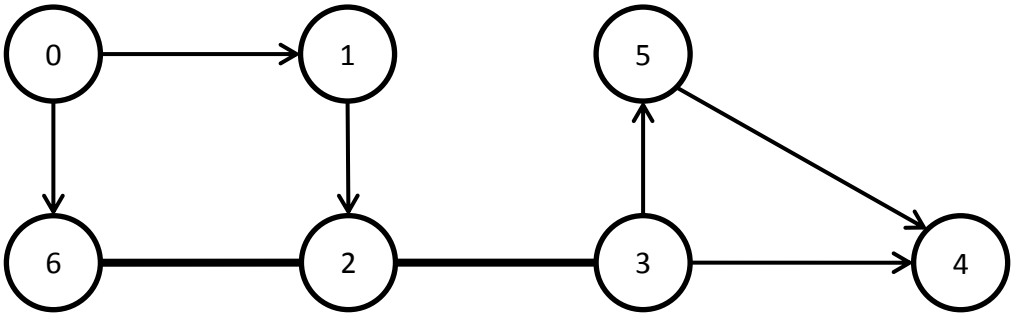
K P B M C N

Teiknaðu LLRB tré sem út kemur. / *Draw the resulting LLRB tree.*

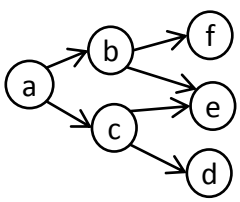
VIII. (8%) Áttuð net/ *Digraphs*

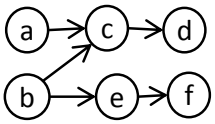
Í áttuðu netunum í þessari spurningu er ekki þekkt hvernig grannlistarnir í netinu eru raðaðir.
/ *In the digraphs of this question, it is not known how the adjacency lists are ordered.*

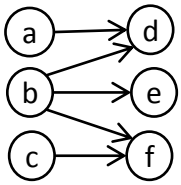
(a) (4%) Áttun tveggja leggja (sýndir með þykkum línum) í netinu að neðan vantar. Gefðu áttun á leggina (með því að teikna á myndina) þannig að DFS postorder röð netsins geti orðið (samkvæmt einhverri röð á grannlistunum): **4 5 3 6 2 1 0.** / *Consider the following digraph. Two of the edges (shown with thick lines) of the digraph below are missing directions. Direct those edges (by drawing on the figure) so that the DFS postorder of the graph can become (under some ordering of the adjacency lists):* **4 5 3 6 2 1 0.**

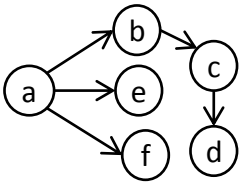


(b) (4%) Í áttuðu netunum að neðan er líka óþekkt hver röð hnútanna er (og þar með hvern hnút DFS heimsækir fyrst). Við hvaða net gæti DFS grannröðunin **a b c d e f** átt (undir einhverri röðun á hnútunum og grannlistunum)? Fleiri en ein gætu verið möguleg. /
In the digraphs below, the ordering of the nodes is also unknown (and therefore also which node is first visited by DFS). To which of the digraphs could the DFS topsort **a b c d e f** *correspond (under some ordering of the nodes and the adjacency lists)? More than one could apply.*





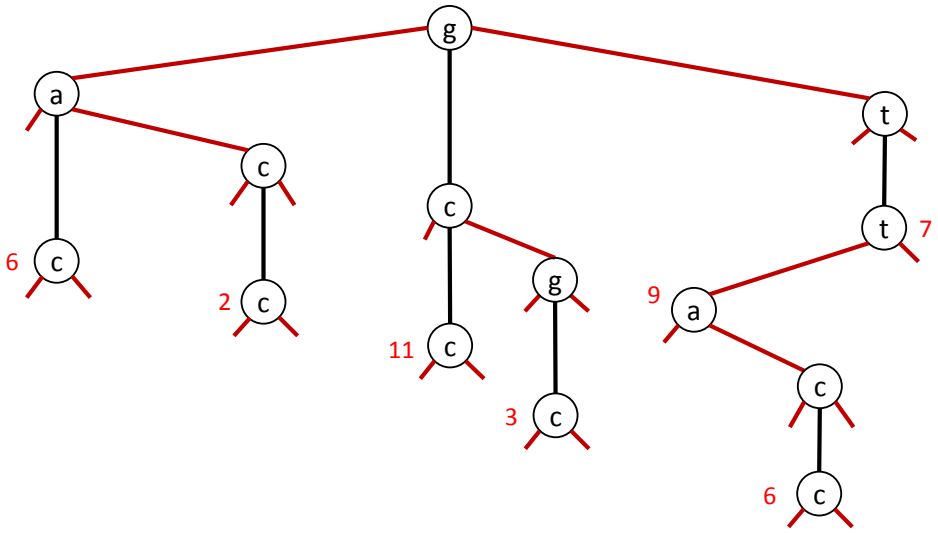




Satt/True:	☐	☐	☐	☐
Ósatt/False:	☐	☐	☐	☐

IX. (8%) Ternary Search Tries

Skoðið eftirfarandi Ternary Search Trie (TST), þar sem gildin eru sýnd við hliðina á nóðunum með samsvarandi strenglykil. / *Consider the following Ternary Search Trie (TST), where the values are shown next to the nodes of the corresponding string keys.*



(a) (3%) Teldu upp lyklana í træinu, í stafrófsröð. */List the keys stored in the trie, in alphabetical order:*

_____ _____ _____ _____ _____ _____ _____

(b) (3%) Gefðu eina mögulega röðun á hvernig lyklarnir gætu hafa verið settir inn í træið.
/ Give one possible order in which the keys were inserted into the trie:

_____ _____ _____ _____ _____ _____ _____

(c) (2%) Stingið inn lyklinum `tag` með gildið 5. Sýnið á myndinni. */ Insert the key* `tag` *into the trie, with value 5. Show on the figure.*

X. (5%) Gagnaþjöppun/*Data compression*

Þú færð eftirfarandi skilaboð sem kóðað var með LZW þjöppun. */ Suppose you receive the following message that was encoded using LZW compression:*

41 81 42 82 83 80

Ljúktu við að afkóða skilaboðin, með einn staf í hverjum ferning. / *Finish decoding the message, writing one letter in each square.*

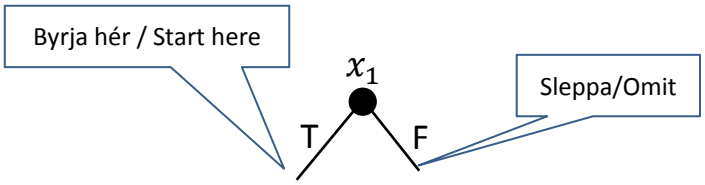
A								

XI. (10%) Satisfiability

Gefin er eftirfarandi CNF boolsk segð: / *We are given the following CNF boolean formula:*

$$x_1x_4x_6,\ \overline{x_1}\,\overline{x_2},\ \overline{x_1}x_3,\ x_2\overline{x_3},\ x_2\overline{x_5},\ x_2\overline{x_7},\ \overline{x_2}\,\overline{x_4},\ \overline{x_2}\,\overline{x_6},\ \overline{x_4}x_6,\ \overline{x_6}x_7$$

(a) (5%) Teiknaðu þann hlutaf af leitartrénu sem *backtracking* aðferð myndar fyrir tilvikið $x_1 = T$. Merktu lauf með þeirri klausu sem myndar mótsögn. / *Draw the portion of the backtracking search tree formed by the case $x_1 = T$. Label each leaf with the clause that forms a contradiction.*



(b) (5%) Teiknaðu nú allt leitartréð sem DPLL aðferðin myndar. Sýndu á hverjum legg þær eindaklausur (unit clauses) sem leiddar eru út, ásamt því að merkja lauf með mótsagnarklausum á sama hátt og áður. / *Now give the whole search tree formed by the DPLL algorithm. Label each edge with the unit clauses derived, as well as mark the leaves with contradicting clauses as before.*

XII. (8%) Satt-Ósatt/True-False

	Satt/True	Ósatt/False
♦ Versta-falls tímaflækja find-aðgerðar með `WeightedQuickUnionUF` á 10^9 stök er í mesta lagi 40. / *The worst-case complexity of a find operation in a `WeightedQuickUnionUF` on 10^9 elements is less than 40.*	☐	☐
♦ Mergesort ber stundum saman sama parið af stökum oftar en einu sinni. / *Mergesort sometimes compare the same pair of elements more than once.*	☐	☐
♦ Ekki er vitað hvort hægt sé að raða hvaða N stökum sem er með $O(N)$ samanburðum. */ It is unknown whether it is possible to sort any N elements in $O(N)$ comparisons*.	☐	☐
♦ InsertionSort er hraðvirkari en Quicksort á fylki með 8 stökum. / *InsertionSort is faster than Quicksort on arrays with 8 elements.*	☐	☐
♦ Raðaðar aðgerðir (eins og að finna minnsta stak) taka $O(\log N)$ tíma þegar notuð er tætitafla með *linear probing*, ef gert er ráð fyrir *uniform hashing assumption*. */ Ordered operations (like finding the minimum) run in $O(\log N)$ time when hashing with linear probing, assuming the uniform hashing assumption.*	☐	☐
♦ Balanseruð tvíleitartré líkt og rauð-svört tré eru fræðilega öflug en ekki notuð í kerfum eins og Java-safninu. / *Balanced binary trees like Red-Black trees are theoretically powerful, but are not used in systems like the Java library.*	☐	☐
♦ Lögun á tvíleitartré fer ekki bara eftir því hvaða stök eru sett í það heldur líka í hvaða röð það er gert. / *The shape of a BST depends not only on which elements are inserted but also on the order in which they are inserted.*	☐	☐
♦ Þegar N stökum er ýtt á stafla sem útfærður er með stækkanlegu (*resizable*) fylki, þá er heildar tímaflækjan $O(N)$. / *When pushing N elements onto a `Stack` implemented as a resizable array, the total time complexity is $O(N)$.*	☐	☐
♦ Það að tvöfalda vægið á hverjum legg í vegnu neti breytir ekki minnsta spanntré þess. / *Doubling the weight of every edge doesn't change the minimum spanning tree of a weighted graph.*	☐	☐
♦ Leggurinn með næstminnsta vægið (þegar allir leggir hafa mismunandi vægi) er ávallt í minnsta spanntrénu. / *The second smallest-weight edge (when all edge-weights are distinct) is always contained in a minimum spanning tree.*	☐	☐
♦ Ef $\mathrm{P} \neq \mathrm{NP}$, þá er Satisfiability leysanlegt á margliðutíma. / *If $P \neq NP$, then Satisfiability is polynomial-time solvable.*	☐	☐
♦ Verkefnið að sannreyna að áttað net sé órásað er í NP. / *The problem of testing if a digraph is acyclic is in NP.*	☐	☐

XIII. (10%) Problem Solving

A. (3%) Þér hefur tekið að fá allar upplýsingar um vinatengsl í félagsnetinu Basehook (Bh). Þú vilt framsenda skilaboð frá einni persónu til annarrar, en hefur tekið eftir því að áhrif skilaboðanna fara mikið eftir því hve mörg vinatengsl þarf að nýta við framsendinguna. Þú vilt sem sagt finna út eftirfarandi: Gefin runa af vinatengslum, og par af ákveðnum persónum A og B, hvað er minnsta gildi k þannig að til sé runa af persónum $X_0, X_1, X_2, \ldots, X_k$ þar sem $A = X_0$, $B = X_k$, og X_i and X_{i+1} eru Bh-vinir, fyrir öll $i = 0,1, \ldots, k-1$. Lýstu stuttlega lausn á þessu verkefni í orðum, með hugtökum námskeiðsins.

Góð lausn er: a) rétt, b) með góða tímaflækju, c) skýr, og d) skorinorð. Gefið er fyrir samkvæmt þessum eiginleikum (í röð eftir mikilvægi).

You have managed to acquire all the informations about friendship connections in the social network Basehook (Bh). You want to forward a message from one person to another, but have observed that the impact of a message depends a lot on along how many connections it has to travel. You therefore want to compute the following: Given a sequence of friendship connections and specific persons A and B, determine the smallest k such that there is a sequence of persons $X_0, X_1, X_2, \ldots, X_k$ where $A = X_0$, $B = X_k$, and X_i and X_{i+1} are Bh-friends, for $i = 0,1, \ldots, k-1$. Describe briefly a solution in words to this problem, using the concepts of this course.

Solutions are graded according to the following criteria (in order of importance): a) correctness, b) time complexity, c) clarity, and d) succinctness.

B. (1%) Hver er versta-falls tímaflækja lausnar þinnar, sem fall af P (fjölda notenda Bh), og F, fjölda vinatengsla í Bh. / *What is the worst-case time complexity of your solution, in terms of P, the number of persons in Bh, and F, the number of friendship connections in* Bh.

Svar: _____

C. (5%) Þú hefur kannað málið frekar og fundið að ekki eru öll vinatengsl jafngild. *Sterk vinatengsl* er öðruvísi, og það kostar ekkert að nota þau: skilaboðin má senda eftir eins mörgum sterkum tengslum eins og verða vill án þess að áhrif skilaboðanna dvíni. Önnur tengsl milli persóna köllum við veik. Þú vilt finna minnsta fjölda veikra tengsla sem þarf til að koma skilaboðum frá A til B (sem gæti nýtt mörg sterk tengsl á leiðinni).

Formlega séð, þá er inntakið runa af þrenndum, `x y z`, þar sem `x` og `y` eru vinir í Bh og `z` er annað hvort `Strong` eða `Weak`, eftir styrk tengslanna. Til viðbótar eru gefnar persónur A og B. Við viljum finna minnsta gildi k þannig að til sé runa af vinatengslum sem tengi A og B, þar sem k þeirra séu `Weak` (og restin `Strong`). Sem dæmi ef það eru sterk tengsl A-C, D-B, og veik tengsl A-E, B-E, C-D, þá er best að senda boðin frá A til C til D til B, því það notar aðeins ein veik tengsl.

Lýstu lausn á þessu verkefni í orðum. Sömu viðmið gilda og í lið A.

You have studied this further and discovered that not all friendships are equal. Strong friendship connections are special, and using such friendship connections doesn't cost anything so to speak: the message can travel along arbitrarily many strong friendships connections without its impact becoming weaker. We call other connections weak. You want to compute the fewest number of weak friendship connections needed to route a message from given person A to person B (possibly using many strong connections).

Formally, the input is a sequency of triples `x y z`*, where* `x` *and* `y` *are* Bh *friends and* `z` *is either* `Strong` *or* `Weak`*, depending on the strength of the relationship. Additionally, we are given specific persons A and B. We want to find the smallest* k *such that there is a chain of friendships connecting A and B, where* k *of those friendships are* `Weak` *(and the rest* `Strong`*). As an example, if there are strong connections A-C, B-D, and weak connections* A-E, B-E, C-D, then it is best to send the message from A to C to D to B, since that uses only one weak connection.

Describe a solution to this problem in words. The same criteria holds as in part A.

XIII. c) frh.

D. (1%) Hver er versta-falls tímaflækja lausnar þinnar, sem fall af P (fjölda notenda Bh), og F, heildarfjölda vinatengsla í Bh. / *What is the worst-case time complexity of your solution, in terms of P, the number of persons in Bh, and F, the total number of friendship connections in* Bh.

Svar: _____