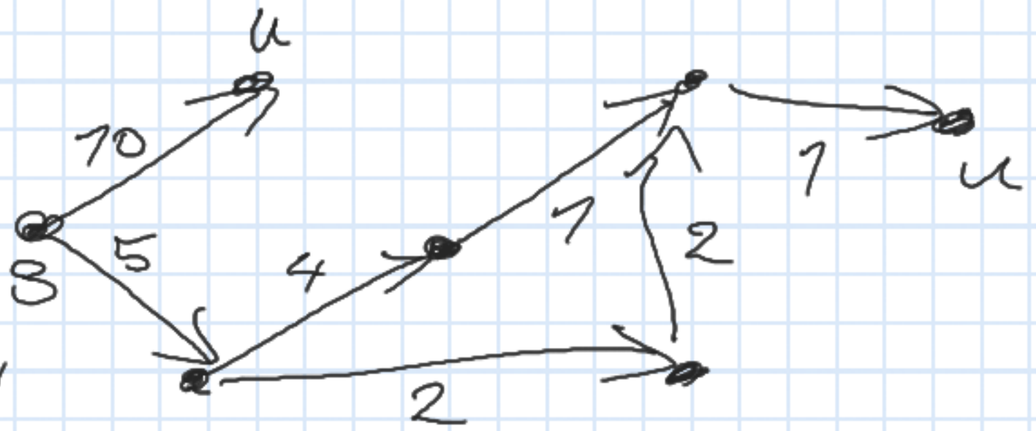
Bit Algo START

Zadanie 5.

Podaj jak najszybszy algorytm obliczający najdłuższą ścieżkę w ważonym acyklicznym grafie skierowanym. Dany jest wierzchołek startowy.



$f(v)$ — dł. najdł. ści. zaczynającej się w v

$f(u) = 0$

$g(i)$ — najd. dr. z s do i

$(u,i) \notin E$

$g(i) = \max_{(u,i)\in E} \{ g(u) + w(u,i) \}$

$f(v) = \max_{(v,u)\in E} \{ w(v,u) + f(u) \}$

Zadanie 1 (egzamin 2020, I termin)

Asystent znanego profesora otrzymał polecenie wyliczenia sumy pewnego ciągu liczb (liczby mogą być zarówno dodatnie jak i ujemne). Aby zminimalizować błędy zaokrągleń asystent postanowił wykonać powyższe dodawania w takiej kolejności, by największy co do wartości bezwzględnej wynik tymczasowy (wynik każdej operacji dodawania; wartość końcowej sumy również traktujemy jak wynik tymczasowy) był możliwie jak najmniejszy.

Aby ułatwić sobie zadanie, asystent nie zmienia kolejności liczb w sumie a jedynie wybiera kolejność dodawań. Napisz funkcję opt sum, która przyjmuje tablicę liczb n1, n2, . . . , nk (w kolejności w jakiej występują w sumie; zakładamy, że tablica zawiera co najmniej dwie liczby) i zwraca największą wartość bezwzględną wyniku tymczasowego w optymalnej kolejności dodawań.

Na przykład dla tablicy wejściowej: [1,−5, 2] funkcja powinna zwrócić wartość 3, co odpowiada dodaniu −5 i 2 a następnie dodaniu 1 do wyniku. Uzasadnij poprawność zaproponowanego rozwiązania i oszacuj jego złożoność obliczeniową.

$[1 + (-5)] + 2 = -4 + 2$

$1 + [(-5) + 2] = 1 + (-3)$

$f(i,j)$ – największy wynik tymczasowy z dodawania elem. od i-th do j-tka

$f(i,i+1) = abs(a[i] + a[i+1])$

$f(i,j) = \max\{abs(sum(i,j)), \min(f(i+1,j), f(i,j-1))\}$

$j > i+1$

$f(i,j) = undef$

$i > j$

	1	2	3	4
10 - 3 + 2 = 9	1	3	6	10

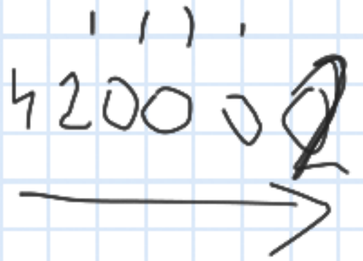
Zadanie 2.

Mamy dany ciąg napisów (słów) $S = [s_1, ..., s_n]$ oraz pewien napis t. Wiadomo, że t można zapisać jako złączenie pewnej ilości napisów z S (z powtórzeniami). Na przykład dla $S = [s_1, s_2, s_3, s_4, s_5]$ gdzie s_1 = ab, s_2 = abab, s_3 = ba oraz s_4 = bab, s_5 = b, napis t = ababbab można zapisać, między innymi, jako s_2s_4 lub jako $s_1s_1s_3s_5$.

Taki wybór konkretnych s_i nazywamy **reprezentacją**. Przez **szerokość** reprezentacji rozumiemy długość najkrótszego s_i należącego do reprezentacji - dla s_2s_4 szerokość to 3, a dla $s_1s_1s_3s_5$ szerokość to 1.

Zaimplementuj algorytm, który mając na wejściu S oraz t znajdzie maksymalną szerokość reprezentacji t (tzn. najkrótszy napis w jej reprezentacji jest najdłuższy). Oszacuj czas działania algorytmu.

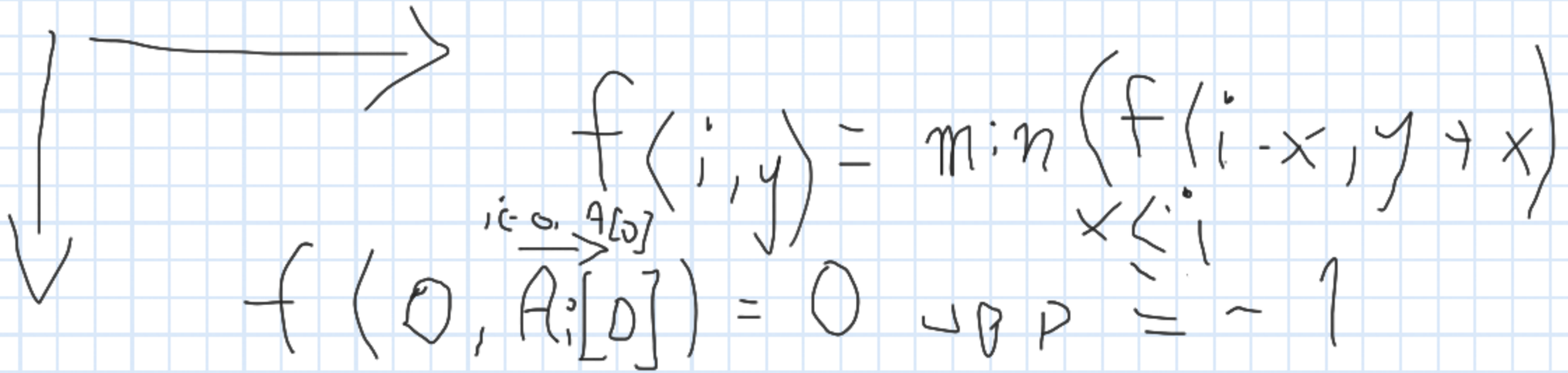
Zadanie 3 (egzamin 2020, II termin)



Żab Zbigniew skacze po osi liczbowej. Ma się dostać z zera do n − 1, skacząc wyłącznie w kierunku większych liczb. Skok z liczby i do liczby j (j > i) kosztuje Zbigniewa j − i jednostek energii, a jego energia nigdy nie może spaść poniżej zera. Na początku Zbigniew ma 0 jednostek energii, ale na szczęście na niektórych liczbach—także na zerze—leżą przekąski o określonej wartości energetycznej (wartość przekąski dodaje się do aktualnej energii Zbigniewa).

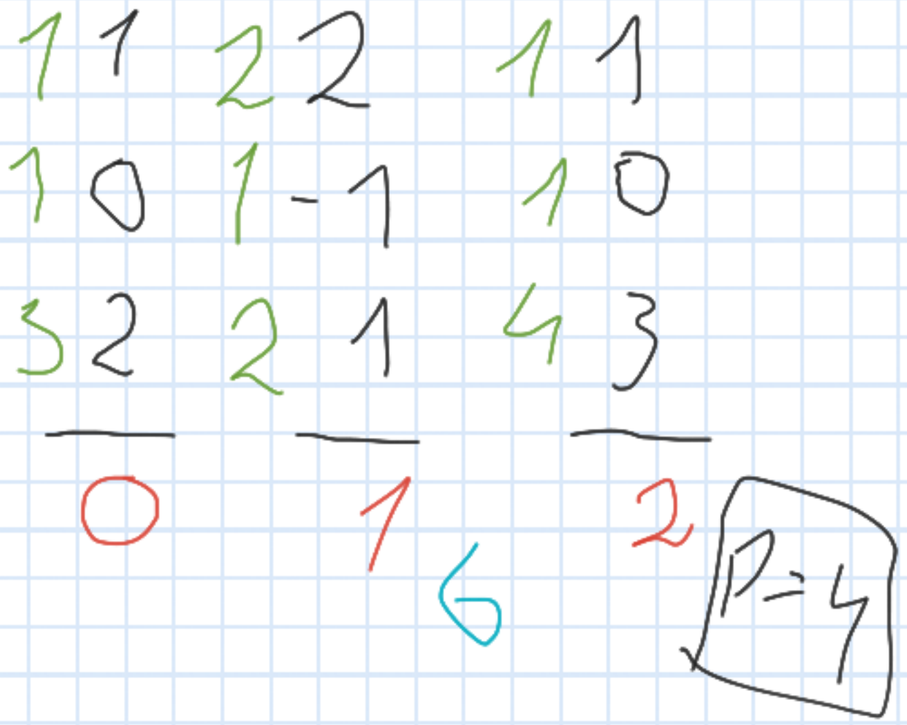
Proszę zaimplementować funkcję zbigniew(A), która otrzymuje na wejściu tablicę A długości len(A) = n, gdzie każde pole zawiera wartość energetyczną przekąski leżącej na odpowiedniej liczbie. Funkcja powinna zwrócić minimalną liczbę skoków potrzebną, żeby Zbigniew dotarł z zera do n-1 lub −1 jeśli nie jest to możliwe.

Podpowiedź. Warto rozważyć funkcję f(i, y) zwracającą minimalną liczbę skoków potrzebną by dotrzeć do liczby i mieć w zapasie dokładnie y jednostek energii.
Przykład. Dla tablicy A = [2,2,1,0,0,0] wynikiem jest 3 (Zbigniew skacze z 0 na 1, z 1 na 2 i z 2 na 5, kończąc z zerową energią). Dla tablicy A = [4,5,2,4,1,2,1,0] wynikiem jest 2 (Zbigniew skacze z 0 na 3 i z 3 na 7, kończąc z jedną jednostką energii).

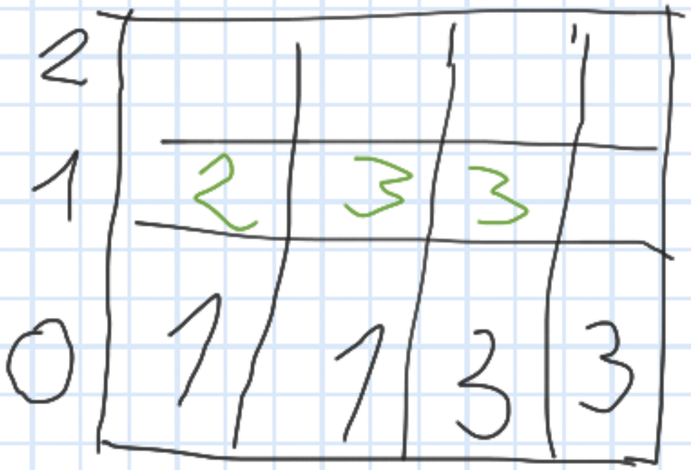


Zadanie 4.

Czarodziej Pascal ma N stosów porcelanowych talerzy, przy czym każdy stos zawiera dokładnie k talerzy. Pascal wystawia dziś wieczorem kolację dla P gości i jedzenie będzie serwowane na tych właśnie talerzach. Każdy talerz ma pewne piękno określone liczbą całkowitą. Pomóż czarodziejowi wybrać dokładnie P talerzy tak, aby miały one maksymalne możliwe piękno. Ale uwaga! Stos to stos, więc jeśli chcesz zabrać jakiś talerz, to musisz też zabrać wszystkie nad nim.



stosy

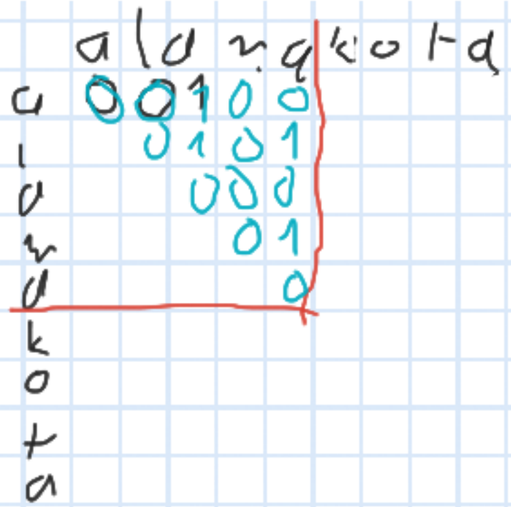


1 2 3 4 => t. tal

$$f(s,t) = \max_{i \in (0,t)} \left(f(s-1, t-i) + P(s,i) \right)$$

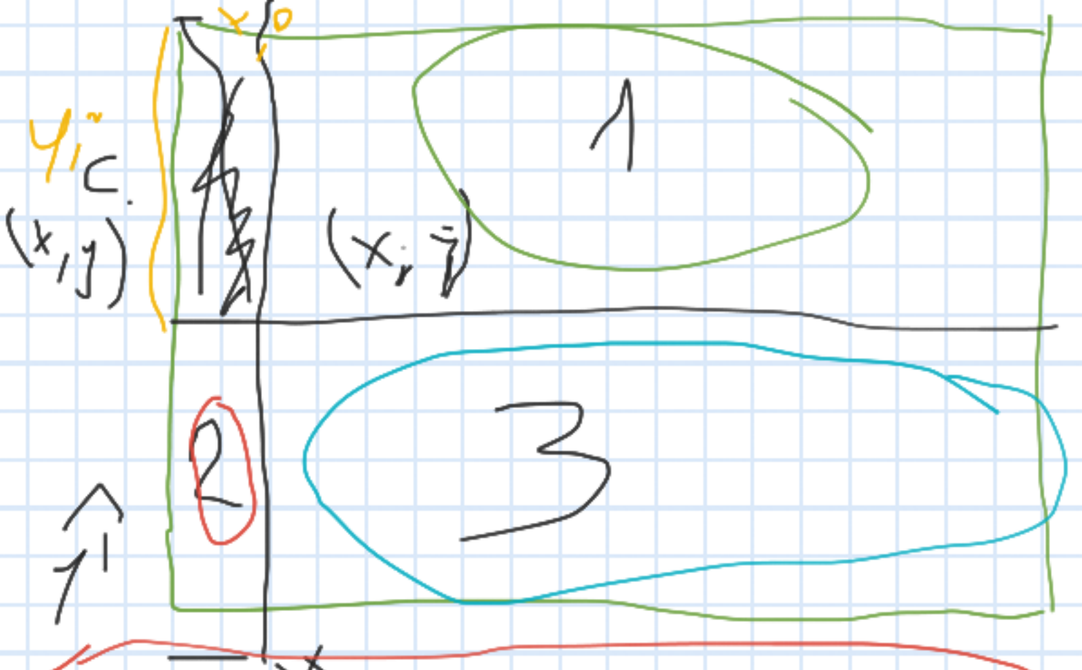
Zadanie 6.

Dana jest zawsze działająca w czasie O(1) funkcja dict(word), która mówi, czy słowo word jest poprawnym słowem danego języka. Dostajemy na wejściu stringa bez spacji. Podaj algorytm, który stwierdzi, czy da się tak powstawiać spacje do wejściowego stringa, że ciąg słów który otrzymamy tworzą słowa z danego języka. Np. “alamakotainiemapsa” możemy zapisać jako “ala ma kota i nie ma psa”. Podaj również, jak wykorzystać algorytm, aby uzyskać przykładowe poprawne rozdzielenie stringa spacjami, jeśli oczywiście ono istnieje. Algorytm ma być szybki, ale najważniejsze, żeby był poprawny!!!.



Zadanie 7.

Dany jest prostokątny kawałek tkaniny o wymiarach X na Y, oraz lista n produktów, które mogą zostać wykonane z tej tkaniny. Do wytworzenia każdego produktu i (i zmienia się od 1 do n) potrzebny jest prostokątny kawałek tkaniny o wymiarach ai na bi, a zysk ze sprzedaży tego produktu wynosi ci. Zakładamy, że ai, bi, ci są dodatnie całkowite. Mając maszynę, która może przeciąć dowolny prostokątny kawałek tkaniny na dwa kawałki wzdłuż linii poziomej lub pionowej, zaprojektuj algorytm, który wyznaczy taką strategię cięcia materiału o wymiarach X na Y, aby sprzedaż wytworzonych produktów dała łącznie jak największy zysk.



$$f(X,Y) = \max_{i:n} \{ c(x_i, y_i) + \dots \}$$

P_x, P_y

$c[(1,8)\ (2,4)\ \dots]$

$c[c_1, c_2, \dots]$

$$+ \max\left(f(P_x - x_i, P_y, -y_i)\right) + \max\left(f(P_x - x, y)\right) + \max\left(f(P_x, \dots\right.$$