**FOGLIO ESPLICATIVO PER GLI ALGORITMI UTILIZZATI NELLE FUNZIONI DELL'HOMEWORK 2**

**OBBLIGATORIO PUBBLICATO IN DATA 26/10/2022**

**SPIEGAZIONE DELLA FUNZIONE "ValoreAssoluto" :**

Prende in input un valore, se minore di zero ritorna il suo corrispettivo moltiplicato per -1,rendendolo

Positivo, altrimenti ritorna il valore originale preso in input.

**SPIEGAZIONE DELLA FUNZIONE "** **xkcd\_to\_list\_of\_weights" :**

Lo scopo della funzione è quello di ricevere in input una stringa che descrive un numero nella codifica

XKCD, per poi restituire all’utente lista di interi, ciascuno corrispondente al peso di una lettera romana .

all’inizio Della funzione, vengono create 2 variabili fondamentali :

“listaInteri” è la variabile che conterrà la lista con tutte le cifre separate e convertite in

intero del numero XKCD

“TenXmultiplayer” Terrà conto del numero di 0 prima di una cifra (controllando da destra verso

sinistra), serve per la conversione dei multipli di 10, durante la spiegazione della task verrà

chiarito meglio il suo funzionamento, viene inizializzata con il valore intero di 1.

Nella prima task il programma inizia iterando per ogni carattere della stringa contenente

Il numero in codifica XKCD, partendo da destra verso sinistra, dato che nella codifica

XKCD, più una cifra è grande/significativa, più si troverà a sinistra.

Per ogni cifra il programma controlla se essa è uguale a 0 o no, ciò è fondamentale.

Sappiamo che nella codifica XKCD gli unici valori composti da più di un carattere

Sono “10”,”100” e “1000”.

Tornando a noi, se la cifra appena controllata NON è uguale a 0, il programma aggiungerà

La cifra convertita ad intero alla “listaInteri” precedentemente inizializzata. La cifra però,

verrà aggiunta moltiplicata alla variabile “TenXmultiplayer” (che, come già detto viene inizializzata

con il valore intero di 1). Dopo essere stata aggiunta, la variabile “TenXmultiplayer” viene

settata nuovamente con il valore di 1, dopo ciò il ciclo è finito e si ripete nuovamente per la prossima

cifra della stringa.

Se la cifra della stringa dovesse essere uguale a 0, il programma piuttosto che aggiungerla alla

“listaInteri”, moltiplicherà il valore di “TenXmultiplayer” per 10. Così che per ogni 0 successivo,

finchè la cifra non sarà 1 (si ricordi che gli unici valori composti da più di un carattere

Sono “10”,”100” e “1000”, quindi per forza di cose dopo una cifra “0” ci sarà o una cifra “0”

O una cifra “1”) il “TenXmultiplayer” verrà incrementato, facendo si che, durante l’iterazione,

quando si arriverà alla cifra “1”, essa verrà aggiunta alla “listaInteri”, moltiplicata per “TenXmultiplayer”,

facendolo diventare il valore giusto per la conversione da XKCD ad intero, resettando poi

“TenXmultiplayer” ad 1 per le cifre successive.

Una volta finito, abbiamo ottenuto una lista di valori interi, che verrà restituita all’utente, dopo esser

Stata invertita nell’ordine corretto tramite la funzione reversed() di python.

**SPIEGAZIONE DELLA FUNZIONE "** **list\_of\_weights\_to\_number " :**

La funzione Trasforma una lista di 'pesi'(numeri interi) nel corrispondente valore arabo, prende in input quindi una lista di valori interi, che chiameremo “listaInteri”.

Si occuperà di capire quali dei valori della “listaInteri” , in base alla loro posizione, dovranno assumere un valore negativo.

Si comincia con un ciclo che verrà ripetuto per ogni elemento della “listaInteri”(che verrà invertita tramite la funzione reversed() di python) tramite La funzione “range” di python.

Abbiamo invertito la lista, quindi essa ha salvato i valori nell’ordine della codifica XKCD, cioè inverso

al nostro sistema decimale, quindi il primo valore della lista, sarebbe l’ultimo carattere

della stringa.

Considerando ciò, nella nostra iterazione controlleremo come prima cosa che non siamo

Alla prima lettura del blocco di codice che sta essendo ripetuto nel ciclo, dato che

Dobbiamo eseguire la task per ogni elemento della lista escluso il primo.

Una volta che ci siamo accertati di non essere al primo ciclo, eseguiremo un controllo:

Tale controllo consiste nell’accertarsi che la cifra che stiamo controllando, sia inferiore

al valore assoluto della cifra che abbiamo controllato nel ciclo precedente.

Se tale accertamento dovesse rivelarsi vero, vuol dire che nella codifica XKCD, c’è una cifra

Seguita da un valore più grande di essa, va quindi sottratta al totale e non aggiunta. Modificheremo

Quindi l’elemento della lista che si riferisce a quella cifra, moltiplicandola per -1, ottenendo

Così il suo corrispettivo negativo. Dato che stiamo modificando in modo dinamico la lista,

durante l’iterazione il fatto che un valore sia stato settato nel suo corrispettivo negativo,

possono verificarsi ovvi problemi nel controllo dove ci accertiamo che la cifra che stiamo controllando,

sia inferiore al valore della cifra che abbiamo controllato nel ciclo precedente.

Dicendo ciò giustifico l’utilizzo della funzione “ValoreAssoluto” creata in precedenza, che mi

Permette di controllare se la cifra che stiamo controllando, sia inferiore al valore assoluto, appunto

reso sempre positivo, della cifra che abbiamo controllato nel ciclo precedente.

Una volta che abbiamo la lista con i valori corretti (cioè negativi quando vanno sottratti al totale),

ritorniamo la somma di tutti i valori della lista tramite la funzione “sum” di python.

**SPIEGAZIONE DELLA FUNZIONE "** **decode\_value " :**

questa funzione decodifica un valore nel formato XKCD e torna l'intero corrispondente. Prende quindi in input una stringa che descrive il suddetto valore nel formato XKCD. Per questa funzione non ho dovuto scrivere codice aggiuntivo, dato che è semplicemente il risultato dell’unione delle due funzioni

“xkcd\_to\_list\_of\_weights” e “xkcd\_to\_list\_of\_weights”.

Difatti, come prima cosa creo una “listaInteri”,

il quale risultato equivale alla funzione xkcd\_to\_list\_of\_weights, alla quale passo come parametro la stringa che descrive il valore nel formato XKCD, ottenuto una lista di valori interi, poi ritorno all’utente il risultato della funzione list\_of\_weights\_to\_number, alla quale passo come parametro “listaInteri” che equivale al corrispondente arabo del valore iniziale passato in input.

**SPIEGAZIONE DELLA FUNZIONE “decode\_XKCD\_tuple”:**

La funzione riceve come input Riceve una lista di stringhe che rappresentano

numeri nel formato XKCD chiamata “xkcd\_values” ed un intero k positivo.

Come prima cosa creo una lista chiamata “listaInteri”, itero per ogni valore

Della lista “xkcd\_values”, e per ognuno di essi eseguo la funzione “decode\_value”

Che ho precedentemente definito (e descritto in questo file di testo) e li inserisco

Nella “listaInteri”, ottenendo una lista con tutti i valori di “xkcd\_values” convertiti

In intero. Dopo di ciò eseguo due iterazioni annidate, in entrambe itero per ogni

Valore della “listaInteri”, facendo si che ogni valore venga messo a confronto con

Tutti gli altri, se un valore è più grande del valore con la quale sta essendo confrontato,

essi vengono scambiati di posto, facendo si che la lista venga ordinata in ordine

crescente.

Ritorno come output il risultato di un'altra iterazione, che parte dall’ultimo valore della lista,

tornando indietro di k posizioni, così da ritornare la lista in ordine decrescente di k elementi.