Laboratorio di Algoritmi

Progetto "Linguistica" (luglio 2023)

Nota: La scadenza del progetto è fissata per lunedì 10 luglio **compreso**.

Nota: Si consiglia di consultare sulla pagina web il documento che riporta le avvertenze utili per lo svolgimento del progetto. Si consiglia anche di verificare di tanto in tanto gli aggiornamenti a questo documento, che potranno riportare risposte ai dubbi degli studenti e correzioni di eventuali errori.

Il problema Si vuole analizzare la struttura di un testo a livello di relazioni fra le parole che lo compongono. In particolare, interessano le frequenze delle parole stesse, e le loro reciproche "distanze". L'idea è che le parole più frequenti racchiudano un'idea del tono generale del testo, e che la vicinanza o lontananza fra l'occorrere delle parole nel testo esprima relazioni fra i concetti che esse esprimono.

Il progetto Il programma da realizzare deve caricare da un file di testo le parole in esso contenute. Le righe del file di testo hanno al massimo 256 caratteri, mentre il numero di righe non è limitato a priori.

Il testo contiene segni di punteggiatura che vanno semplicemente ignorati cancellandoli. Per semplicità, ipotizzeremo che qualunque carattere non sia una lettera dell'alfabeto sia un segno di punteggiatura e vada ignorato, anche se questo potrebbe scomporre una parola singola in più parole. Per esempio, la parola singola "`forty-nine`" diventerà "`forty nine`", spezzandosi in due parole distinte).

Si ricorda che la funzione `fscanf` con la specifica `"%s"` interrompe la lettura solo quando incontra caratteri separatori. Questa difficoltà ha molte soluzioni semplici. Una è usare le specifiche `"%[a-zA-Z]"` e `"%[^a-zA-Z]"` alternativamente per leggere sequenze di caratteri alfabetiche ("parole") e non alfabetiche ("non-parole"). Un'altra è leggere il file una riga alla volta, metterla in una stringa, "ripulire" quest'ultima e procedere a riconoscerne le parole. Sicuramente vi sono altre soluzioni. Siete pregati di evitare soluzioni complicate.

Elaborando il testo, ignoreremo le differenze fra lettere maiuscole e minuscole, adottando la forma minuscola come standard (per esempio, `THEY`, `They` e `they` sono tutte occorrenze della parola `they`). Ignoreremo anche il fatto che la stessa parola può assumere forme leggermente diverse a seconda che sia singolare o plurale, ecc. . . : le considereremo quindi come parole distinte (per esempio, `project` e `projects` sono parole diverse). D'altra parte, considereremo parole di significato diverso, ma di uguale forma, come occorrenze della stessa parola (per esempio, `can` come voce del verbo "potere" e `can` come "barattolo").

Per esempio, si consideri il testo seguente:

```
Sopra la panca, la capra campa;
sotto la panca, la capra crepa.
```

Ignorando i segni di punteggiatura e lettere maiuscole il testo diventa:

```
sopra la panca la capra campa sotto la panca la capra crepa
```

Il programma deve per prima cosa stampare una riga con il numero di parole diverse contenute nel testo, seguito dalla parola chiave `parole` e un'altra riga con il numero totale di occorrenze, seguito dalla parola chiave `occorrenze`. Nell'esempio considerato:

```
7 parole
12 occorrenze
```

Il programma deve determinare le $K = 20$ parole più frequenti nel testo e stamparle, una per riga, seguite dalla loro frequenze assolute (numero di occorrenze nel testo), ordinate per valori decrescenti. A parità di frequenza, il programma stampa le parole in ordine alfabetico. Se le parole sono meno di K, il programma le stampa tutte. Nell'esempio:

```
la 4
capra 2
panca 2
campa 1
crepa 1
sopra 1
sotto 1
```

dato che le parole sono meno di K, la parola `la` compare 4 volte, e così via.

Definiremo poi come *distanza* fra due parole la minima differenza fra le posizioni nel testo di un'occorrenza della prima e una della seconda. Per esempio, la parola `panca` compare nelle posizioni 3 e 9 e la parola la parola `campa` in posizione 6. La loro distanza è $\min(|3-6|, |9-6|) = 3$. Vogliamo determinare il numero di coppie (non ordinate) di parole che sono a distanza 1, a distanza 2 e a distanza 3 e stamparle se tre righe indicando in ciascuna il numero, seguito dalle parole chiave `coppie a distanza` e dalla distanza stessa. Nell'esempio:

```
7 coppie a distanza 1
6 coppie a distanza 2
2 coppie a distanza 3
```

dato che le coppie `(campa,capra)`, `(campa,sotto)`, `(capra,crepa)`, `(capra,la)`, `(la,panca)`, `(la,sopra)`, `(la,sotto)` sono adiacenti, e così via.

Ci concentreremo poi sulle coppie di parole a distanza $d(x, y) \leq D = 2$ e costruiremo un grafo avente come vertici tutte le parole e come lati le coppie entro la distanza limite. Stamperemo i lati di questo grafo in ordine lessicografico, cioè ordinandoli in base prima alla prima parola della coppia e poi alla seconda (e ovviamente ordinando le due parole di ciascuna coppia allo stesso modo). Ogni lato occuperà una riga, con le due parole racchiuse fra parentesi e separate da una virgola, seguite dal valore della distanza. Nell'esempio, i 13 lati del grafo (le 7 coppie a distanza 1 e le 6 coppie a distanza 2) saranno:

```
(campa,capra) 1
(campa,la) 2
(campa,sotto) 1
(capra,crepa) 1
(capra,la) 1
(capra,panca) 2
(capra,sotto) 2
(crepa,la) 2
(la,panca) 1
(la,sopra) 1
(la,sotto) 1
(panca,sopra) 2
(panca,sotto) 2
```

A questo punto, ignorando la funzione distanza sopra definita, e considerando invece il numero di lati (che in qualche modo le è legato, ma non equivalente),

calcoleremo per ogni coppia di parole la lunghezza del cammino minimo che li collega. Per due vertici adiacenti, la lunghezza sarà 1; se bisogna necessariamente passare per un terzo vertice, sarà 2, e così via. Si stamperà il valor medio della lunghezza di tali cammini minimi fra tutte le coppie di vertici, arrotondato a due cifre decimali (essendo un numero reale) e preceduto dalle parole chiave `lunghezza media`. In questo calcolo considereremo solo le coppie di vertici diversi: non interessa il cammino minimo fra un vertice e sé stesso. Infine, stamperemo una riga con il valore massimo della lunghezza dei cammini minimi, detto *diametro* del grafo, preceduto dalla parola chiave `diametro`.

Nell'esempio, le lunghezze dei cammini minimi fra coppie di vertici diversi sono:

	campa	capra	crepa	la	panca	sopra	sotto
campa	-	1	2	1	2	2	1
capra	-	-	1	1	1	2	1
crepa	-	-	-	1	2	2	2
la	-	-	-	-	1	1	1
panca	-	-	-	-	-	1	1
sopra	-	-	-	-	-	-	2
sotto	-	-	-	-	-	-	-

Di conseguenza, le coppie sono 21 e la somma delle distanze è 29, per cui $29/21 = 1.38095\ldots$ e il valore massimo è 2. Stamperemo:

```
lunghezza media 1.38
diametro 2
```

Chiarimenti

Come si devono considerare le cifre? Vanno ignorate, non essendo lettere dell'alfabeto?

Sì, tutti i caratteri diversi da lettere, maiuscole o minuscole, vanno ignorati, cioè trattati come spazi bianchi.