Opdracht

Uitgebreide opdrachtbeschrijving is te vinden in Java Magazine 2014, no. 2. Samenvatting: vind een permutatie van ons alfabet die het aantal woorden dat alleen oplopende letters heeft maximaliseert. In ons gebruikelijke alfabet is dan ANNO geldig, maar ADMIN niet. Er wordt een woordenlijst van 67.200 engelse woorden gebruikt die de downloaden is van http://bit.ly/alfabetsoep . Onmogelijke woorden zijn hieruit al verwijderd (bijvoorbeeld NONE).

De beste oplossing

In de opdracht staat dat “het waarschijnlijk af te raden valt alle permutaties af te gaan...”. Als we een alfabet willen dat gegarandeerd het meeste woorden toestaat, ontkomen we daar in het slechtste geval niet aan. Het slechtste geval zal met een onwaarschijnlijk lage kans voorkomen, maar ook het te verwachten aantal permutaties dat beschouwd moet worden voor de beste oplossing is ondoenlijk groot.

Dit komt omdat het verlies aan woorden bij het toevoegen van een letter niet alleen bepaald wordt door het verlies dat die letter met zijn voorgangers veroorzaakt, maar ook door de volgorde waarin deze voorgangers staan. Als woorden die door de relatie van de toegevoegde letter met zijn voorgangers verloren gaan al verloren waren gegaan levert dit immers geen extra verlies op. De ondergrens van verlies door toevoegen van een letter aan een deeloplossing is daarmee 0.

De opgave is voor te stellen als een boom waarvan we het goedkoopste blad moeten vinden. Vanuit de wortel zijn er 26 vertakkingen, vanuit elk van die takken 25 takken, enz. Dat de ondergrens niet hoger is dan 0 zorgt ervoor dat we een pad richting blad pas de deur kunnen wijzen als het verlies groter is dan het verlies van het beste blad wat we tot nu gezien hebben.

In Tabel 1 op de volgende pagina is het verlies per positie op basis van 10.000 willekeurige alfabetten te zien. Gemiddeld zul je tot en met de 23^{e} positie moeten zoeken voor je boven de beste oplossing van 10.000 willekeurige alfabetten zit, wat ondoenlijk is.

Een goede oplossing

Bij gebrek aan exact algoritme gebruiken we het Local Search algoritme Simulated Annealing. Hiermee vinden we niet gegarandeerd de beste oplossing, maar wel vaak goede oplossingen. Simulated Annealing begint met een geldige oplossing en verandert die. Als dit een verbetering is, wordt er vanaf deze oplossing verder gezocht. Als het een verslechtering is wordt er vanaf deze oplossing verder gezocht met een kans die kleiner is als de verslechtering groter is, of als we later in de zoektocht zitten.

Implementatie

Het bepalen van de kosten van een oplossing is de bottleneck in de rekentijd. Het volledig berekenen van een pad kost ongeveer 0.5 * 26^2 set verenigingen (verlies tot nu toe met het verlies van elke letter combinatie).

Het veranderen van de huidige oplossing gebeurt aan de hand van 2 indices ('links' en 'rechts'). Mogelijke veranderingen zijn: verplaatsen van 'links' naar 'rechts', verplaatsen van 'rechts' naar 'links', omwisselen, willekeurige permutatie tussen 'links' en 'rechts'.

Per pad element worden er twee dingen gecached: het verlies van dit element met zijn voorgangers en het totale verlies van het pad tot en met dit element. Met behulp van deze cache hoeft na een verandering op basis van de indices 'links' en 'rechts' niet het hele pad opnieuw berekend te worden. Het pad voor 'links' kan volledig hergebruikt worden. Tussen 'links' en 'rechts' moet opnieuw gerekend worden. Na 'rechts' moet het totale verlies opnieuw berekend worden, maar hierbij kan wel gebruikt gemaakt worden het verlies met zijn voorgangers uit zijn cache.

Op deze manier kost het berekenen van de kosten van het nieuwe pad nog ongeveer ('rechts' – 'links') * ('links' + 0.5 * ('rechts' – 'links')) + 26 – 'rechts' set verenigingen. In het slechtste geval is dit hetzelfde als het gehele pad berekenen, maar meestal is het minder werk.

Resultaat
De beste oplossing die gevonden is, is niet 1 alfabet, maar een set van 85 gelijkwaardige alfabetten. Deze alfabetten blokkeren allen 63.184 van de 67.230 woorden, waarmee ze dus 4.046 geldige woorden over laten. De alfabetten zijn te vinden in Tabel 2 op de volgende pagina's.

Tabel 1
Verlies per positie op basis van 10.000 willekeurige alfabetten

Index	Gemiddeld	Minimum	Maximum
1	0.0	0.0	0.0
2	0.0311	0.0	0.2208
3	0.0829	1.0E-4	0.4416
4	0.1483	0.0012	0.548
5	0.2211	0.0054	0.6598
6	0.2983	0.0166	0.6918
7	0.3759	0.0317	0.7846
8	0.4503	0.0616	0.8107
9	0.5201	0.0892	0.8454
10	0.5859	0.1372	0.8766
11	0.6457	0.1542	0.9086
12	0.7001	0.2376	0.9214
13	0.7476	0.3164	0.9614
14	0.7891	0.3999	0.9639
15	0.8254	0.4518	0.967
16	0.8568	0.5741	0.9694
17	0.8836	0.6114	0.9736
18	0.9064	0.7031	0.9804
19	0.9254	0.7214	0.9833
20	0.9412	0.818	0.9865
21	0.9539	0.8489	0.9895
22	0.9642	0.8691	0.9921
23	0.9728	0.8964	0.9935
24	0.9796	0.9402	0.9943
25	0.9849	0.9549	0.9963
26	0.9891	0.9727	0.9968

Tabel 2
Alfabetten die 4.046 woorden met oplopende letters bevatten

[B, C, F, J, W, H, L, O, A, Q, U, M, V, X, P, I, N, G, T, Z, K, E, R, D, Y, S]
[B, C, F, W, H, L, J, O, A, Q, U, M, X, V, P, I, N, G, T, K, Z, E, R, D, Y, S]
[B, C, F, W, H, L, J, O, A, Q, U, X, M, V, P, I, N, G, T, K, Z, E, R, D, Y, S]
[B, C, F, W, J, H, L, O, A, Q, U, M, V, X, P, I, N, G, T, Z, K, E, R, D, Y, S]
[B, C, F, W, J, H, L, O, A, Q, U, X, M, V, P, I, N, T, K, G, Z, E, R, D, Y, S]
[B, C, P, F, H, J, L, O, W, A, Q, U, M, X, V, I, N, T, K, G, Z, E, R, D, Y, S]
[B, C, P, H, F, J, L, O, W, A, Q, U, X, M, V, I, N, T, Z, K, G, E, R, D, Y, S]
[B, C, P, H, F, L, J, O, W, A, Q, U, X, M, V, I, N, T, G, Z, K, E, R, D, Y, S]
[B, C, W, F, H, L, J, O, A, Q, U, X, M, V, P, I, N, G, T, K, Z, E, R, D, Y, S]
[B, C, W, F, J, H, L, O, A, Q, U, M, V, X, P, I, N, G, T, K, Z, E, R, D, Y, S]
[B, F, C, W, H, L, J, O, A, Q, U, M, P, V, X, I, N, G, T, K, Z, E, R, D, Y, S]
[B, F, C, W, H, L, J, O, A, Q, U, M, P, X, V, I, N, G, T, K, Z, E, R, D, Y, S]
[B, F, C, W, H, L, J, O, A, Q, U, M, X, V, P, I, N, G, T, K, Z, E, R, D, Y, S]
[B, P, C, F, H, J, L, O, W, A, Q, U, M, V, X, I, N, G, K, T, Z, E, R, D, Y, S]
[B, P, C, F, H, J, L, O, W, A, Q, U, X, M, V, I, N, K, T, G, Z, E, R, D, Y, S]
[B, P, C, F, H, J, L, O, W, A, Q, U, X, M, V, I, N, K, T, Z, G, E, R, D, Y, S]
[B, P, C, F, H, J, L, O, W, A, Q, U, X, M, V, I, N, T, G, Z, K, E, R, D, Y, S]
[B, P, C, H, F, L, J, O, W, A, Q, U, X, M, V, I, N, T, G, Z, K, E, R, D, Y, S]
[B, P, F, C, H, J, L, O, W, A, Q, U, M, X, V, I, N, K, T, G, Z, E, R, D, Y, S]
[B, P, F, C, H, J, L, O, W, A, Q, U, X, M, V, I, N, K, T, Z, G, E, R, D, Y, S]
[C, B, F, J, W, H, L, O, A, Q, U, M, X, V, P, I, N, T, G, Z, K, E, R, D, Y, S]

[C, B, F, J, W, H, L, O, A, Q, U, X, M, V, P, I, N, T, G, K, Z, E, R, D, Y, S]
[C, B, F, J, W, H, L, O, A, Q, U, X, M, V, P, I, N, T, G, Z, K, E, R, D, Y, S]
[C, B, F, W, J, H, L, O, A, Q, U, M, X, P, V, I, N, T, Z, G, K, E, R, D, Y, S]
[C, B, F, W, J, H, L, O, A, Q, U, X, M, V, P, I, N, T, K, G, Z, E, R, D, Y, S]
[C, B, P, F, J, H, L, O, W, A, Q, U, M, V, X, I, N, T, G, Z, K, E, R, D, Y, S]
[C, B, P, F, J, H, L, O, W, A, Q, U, M, V, X, I, N, T, Z, G, K, E, R, D, Y, S]
[C, B, P, F, J, H, L, O, W, A, Q, U, M, X, V, I, N, T, Z, G, K, E, R, D, Y, S]
[C, B, P, H, F, J, L, O, W, A, Q, U, X, M, V, I, N, T, Z, K, G, E, R, D, Y, S]
[C, B, P, H, F, L, J, O, W, A, Q, U, X, M, V, I, N, G, T, K, Z, E, R, D, Y, S]
[C, B, P, H, F, L, J, O, W, A, Q, U, X, M, V, I, N, G, T, Z, K, E, R, D, Y, S]
[C, B, W, F, H, L, J, O, A, Q, U, M, V, P, X, I, N, T, G, K, Z, E, R, D, Y, S]
[C, B, W, F, H, L, J, O, A, Q, U, X, M, V, P, I, N, T, K, G, Z, E, R, D, Y, S]
[C, F, B, J, W, H, L, O, A, Q, U, M, V, X, P, I, N, G, T, Z, K, E, R, D, Y, S]
[C, F, B, J, W, H, L, O, A, Q, U, X, M, P, V, I, N, T, Z, K, G, E, R, D, Y, S]
[C, F, B, J, W, H, L, O, A, Q, U, X, M, V, P, I, N, T, G, K, Z, E, R, D, Y, S]
[C, F, B, J, W, H, L, O, A, Q, U, X, M, V, P, I, N, T, Z, K, G, E, R, D, Y, S]
[C, F, B, W, H, L, J, O, A, Q, U, X, M, V, P, I, N, T, K, G, Z, E, R, D, Y, S]
[C, F, B, W, J, H, L, O, A, Q, U, M, V, X, P, I, N, G, T, K, Z, E, R, D, Y, S]
[C, F, B, W, J, H, L, O, A, Q, U, M, V, X, P, I, N, G, T, Z, K, E, R, D, Y, S]
[C, F, B, W, J, H, L, O, A, Q, U, M, X, P, V, I, N, T, Z, G, K, E, R, D, Y, S]
[C, F, B, W, J, H, L, O, A, Q, U, M, X, V, P, I, N, T, G, Z, K, E, R, D, Y, S]
[C, F, B, W, J, H, L, O, A, Q, U, X, M, P, V, I, N, G, T, Z, K, E, R, D, Y, S]
[C, F, B, W, J, H, L, O, A, Q, U, X, M, P, V, I, N, T, Z, K, G, E, R, D, Y, S]
[C, F, J, W, B, H, L, O, A, Q, U, M, V, X, P, I, N, G, T, Z, K, E, R, D, Y, S]
[C, F, J, W, B, H, L, O, A, Q, U, X, M, V, P, I, N, G, T, Z, K, E, R, D, Y, S]
[C, F, J, W, B, H, L, O, A, Q, U, X, M, V, P, I, N, T, G, Z, K, E, R, D, Y, S]
[C, F, W, B, J, H, L, O, A, Q, U, M, V, X, P, I, N, G, T, Z, K, E, R, D, Y, S]
[C, F, W, B, J, H, L, O, A, Q, U, X, M, P, V, I, N, G, T, Z, K, E, R, D, Y, S]
[C, F, W, J, B, H, L, O, A, Q, U, X, M, V, P, I, N, T, Z, K, G, E, R, D, Y, S]
[C, P, F, B, J, H, L, O, W, A, Q, U, M, V, X, I, N, T, G, Z, K, E, R, D, Y, S]
[C, P, F, B, J, H, L, O, W, A, Q, U, M, X, V, I, N, T, G, Z, K, E, R, D, Y, S]
[C, W, B, F, H, L, J, O, A, Q, U, M, V, P, X, I, N, T, Z, G, K, E, R, D, Y, S]
[C, W, B, F, H, L, J, O, A, Q, U, M, X, V, P, I, N, T, G, Z, K, E, R, D, Y, S]
[C, W, B, F, H, L, J, O, A, Q, U, X, M, V, P, I, N, T, G, Z, K, E, R, D, Y, S]
[C, W, B, F, H, L, J, O, A, Q, U, X, M, V, P, I, N, T, K, Z, G, E, R, D, Y, S]
[C, W, B, F, H, L, J, O, A, Q, U, X, M, V, P, I, N, T, Z, G, K, E, R, D, Y, S]
[C, W, B, F, J, H, L, O, A, Q, U, M, X, P, V, I, N, T, Z, G, K, E, R, D, Y, S]
[C, W, B, F, J, H, L, O, A, Q, U, M, X, P, V, I, N, T, Z, K, G, E, R, D, Y, S]
[C, W, B, F, J, H, L, O, A, Q, U, M, X, V, P, I, N, T, G, Z, K, E, R, D, Y, S]
[C, W, F, B, J, H, L, O, A, Q, U, M, X, P, V, I, N, G, T, Z, K, E, R, D, Y, S]
[C, W, F, B, J, H, L, O, A, Q, U, X, M, P, V, I, N, G, T, Z, K, E, R, D, Y, S]
[F, B, C, W, H, J, L, O, A, Q, U, M, X, V, P, I, N, T, G, Z, K, E, R, D, Y, S]
[F, B, C, W, J, H, L, O, A, Q, U, M, P, V, X, I, N, T, G, K, Z, E, R, D, Y, S]
[F, B, C, W, J, H, L, O, A, Q, U, M, V, X, P, I, N, G, T, Z, K, E, R, D, Y, S]
[F, C, B, W, H, J, L, O, A, Q, U, M, P, X, V, I, N, T, G, Z, K, E, R, D, Y, S]
[F, C, B, W, H, J, L, O, A, Q, U, M, P, X, V, I, N, T, K, G, Z, E, R, D, Y, S]
[F, C, B, W, H, J, L, O, A, Q, U, M, X, P, V, I, N, T, G, Z, K, E, R, D, Y, S]
[F, C, J, W, B, H, L, O, A, Q, U, M, V, X, P, I, N, G, T, Z, K, E, R, D, Y, S]
[F, J, B, C, W, H, L, O, A, Q, U, M, X, P, V, I, N, T, G, Z, K, E, R, D, Y, S]
[F, J, C, B, W, H, L, O, A, Q, U, M, X, P, V, I, N, T, G, Z, K, E, R, D, Y, S]
[F, J, C, W, B, H, L, O, A, Q, U, X, M, V, P, I, N, T, G, K, Z, E, R, D, Y, S]
[P, B, C, F, H, J, L, O, W, A, Q, U, X, M, V, I, N, K, T, G, Z, E, R, D, Y, S]
[P, B, C, H, F, J, L, O, W, A, Q, U, M, X, V, I, N, T, Z, G, K, E, R, D, Y, S]
[P, B, C, H, F, J, L, O, W, A, Q, U, X, M, V, I, N, T, Z, G, K, E, R, D, Y, S]
[P, B, C, H, F, J, L, O, W, A, Q, U, X, M, V, I, N, T, Z, K, G, E, R, D, Y, S]
[P, B, C, H, F, L, J, O, W, A, Q, U, M, X, V, I, N, T, Z, G, K, E, R, D, Y, S]
[P, B, C, H, F, L, J, O, W, A, Q, U, X, M, V, I, N, G, T, Z, K, E, R, D, Y, S]
[P, B, C, H, F, L, J, O, W, A, Q, U, X, M, V, I, N, T, G, Z, K, E, R, D, Y, S]
[P, B, F, C, H, J, L, O, W, A, Q, U, M, V, X, I, N, K, T, G, Z, E, R, D, Y, S]
[P, C, B, F, J, H, L, O, W, A, Q, U, M, V, X, I, N, T, G, K, Z, E, R, D, Y, S]
[P, C, B, F, J, H, L, O, W, A, Q, U, M, V, X, I, N, T, Z, K, G, E, R, D, Y, S]
[P, C, B, H, F, L, J, O, W, A, Q, U, X, M, V, I, N, G, T, Z, K, E, R, D, Y, S]

[P, F, C, B, H, J, L, O, W, A, Q, U, X, M, V, I, N, K, T, Z, G, E, R, D, Y, S]
[P, F, J, B, C, H, L, O, W, A, Q, U, X, M, V, I, N, T, Z, G, K, E, R, D, Y, S]