Opracowanie: POC
Pociąg

AUTORZY:
• v. 1.00: 2010-08-20, autor zadania: Jacek Tomasiewicz, opracowanie: Adrian Jaskółka

PROSERWY 2010

1 Wstęp

Niech n będzie liczbą wagonów, m będzie długością tekstu t oraz s będzie sumą długości numerów seryjnych.

2 Rozwiązanie brutalne

Zadanie można w dosyć prosty sposób zrobić w złożoności $O(n! * s * m)$, po prostu przeglądając wszystkie permutacje wagonów, wyszukując wzorca w tekscie brutalnie w złożoności $O(s * m)$.

3 Rozwiązanie wolne

Powyższe rozwiązanie można udoskonalić, wyszukując wzorzec liniowo przy pomocy algorytmu *KMP*, *Rabina-Karpa*, bądź *Boyera-Moore'a*. Opisy tych algorytmów można znaleźć w literaturze bądź w internecie. Rozwiązanie owo, działa w złożoności $O(n! * (s + m))$.

4 Rozwiązanie wzorcowe

Przedstawione algorytmy nie korzystają z faktu, że wszystkie wzorce mają takie same długości. Dla rozwiązania wzorcowego, jest to bardzo istotne. Można bowiem zmodyfikować algorytm *Rabina-Karpa*, tak aby w złożoności $O(w + t * log(t) + l * log(t))$ (gdzie t to długość tekstu, w to suma długości wzorców, a l to liczba wzorców) znaleźć wszystkie wystąpienia.

Dalsza część omówienia, wymaga znajomości algorytmu *Rabina-Karpa* w postaci podstawowej. Aby rozszerzyć ten algorytm, to wpierw znajdujemy hasze wszystkich podsłów tekstu o długości takiej, jaki ma każdy wzorzec, oraz wrzucamy je sobie do mapy M, zliczając ile razy występuje jaki hasz. Następnie wystarczy zhaszować wszystkie wzorce i dla każdego przy pomocy mapy M stwierdzić ile razy występuje on w tekście. Jednak zastowowanie tego algorytmu wprost, da nam złożoność $O(n! * s + m * log(m) + n! * log(m))$, czyli gorzej niż drugie rozwiązanie. Widać że najbardziej czasochłonne jest tutaj liczenie haszy dla wzorców.

Można to zrobić szybciej w złożoności $O(n! * n)$ dla wszystkich wzorców. Korzystamy tutaj z faktu że nasze wzorce są popermutowanymi zlepkami tych samych słów. Dokładny sposób jak to zrobić, zostawię jako ćwiczenie dla Czytelnika. Końcową złożonością naszego rozwiązania jest zatem $O(n!*n+m*log(m)+n!*log(m))$.

5 Dodatkowe uwagi

Hashe słów należy zawsze liczyć, trzymać i generalnie robić z nimi wszystko na long long-ach. Wiąże się to z tzw. *paradoksem dnia urodzin*, który mówi, że jeżeli losujemy liczby z przedziału od 1 do n, to przeciętna liczba losowań potrzebna do tego by jakaś się powtórzyła jest proporcjonalna do $\sqrt{n}$. Wg. badań przeprowadzonych przez amerykańskich naukowców, przeciętna liczba losowań liczb z zakresu int-a, tak aby jakaś się powtórzyła wynosi około 82000. Tak więc haszując na intach, średnio co 82000 różne słowo będzie miało ten sam hasz, co będzie prowadziło do kolizji i błędnej odpowiedzi.

Niestety amerykańscy naukowcy nie dysponowali wystarczająco dobrym sprzętem aby doprowadzić choćby do jednej kolizji na long longach, co jest najlepszym dowodem niezawodności long longów.