Zadanie: PSO
Psotnik Bidek

Warsztaty ILO, grupa olimpijska, dzień 16. Dostępna pamięć: 256 MB.

Rozwiązanie brutalne $O(2^n * 27)$

Założenie o wielkości n i natura zadania od razu nasuwają rozwiązanie korzystające z programowania dynamicznego.

Nasze rozwiązanie będzie obliczało odpowiedź dla każdego zbioru słów, zaczynając od najmniej licznych do pełnego zbioru, który będzie naszym wynikiem (Tzw. dp po maskach bitowych).

Obserwacja.1. *Jeśli do jakiegoś słowa dokładamy kolejną literkę, powiedzmy a, to liczba różnych podciągów rośnie dwukrotnie, odjąć liczba podsłów zawierających a jako ostatnią literę plus 1, jeśli będziemy w to wliczać również słowo puste jako poprawne słowo.*

Dowód. powinno to być dość jasne, wartym komentarza jest fakt skąd bierze się plus 1. Jest to słowo składające się ze wszystkich dotychczasowych wystąpień litery a. □

dodajmy do tego drobną obserwację:

Obserwacja.2. *Zamiast trzymać dokładne wartości wystarczy trzymać jedynie parzystość wszystkich wartości.*

Dowód. jest to dość prosta obserwacja, ponieważ zależy nam jedynie na tym czy liczba różnych podciągów jest parzysta, to wystarczy trzymać resztę z dzielenia tej wartości przez 2. □

Teraz mając te 2 obserwacje prawdopodobnie bylibyśmy w stanie uzyskać rozwiązanie gdzie wymiarem dp jest podzbiór zbioru słów (2^n), oraz parzystości naszego wyniku oraz wyników z założeniem że jest to ostatnia litera, (ponieważ każdą tą wartość trzymamy jako 0 lub 1, to łącznie jest to dodatkowe 2^{27} operacji).

Zajmijmy się teraz samymi słowami, zauważmy że są to funkcje z pewnej maski (2^{27}) w inną maskę (2^{27}), obie reprezentujące parzystości wyniku oraz wyniku z założeniem że słowo musi kończyć się na daną literkę.

Obserwacja.3. *W każdej masce reprezentującej wyniki zawsze dokładnie jedna wartość jest równa 1.*

Dowód. Jest to jedyna obserwacja która nawet po usłyszeniu jej, ciężko zorientować się skąd się bierze. Najłatwiejszym sposobem jest zorientowanie się że słowo puste zawiera 1 jedynie na pozycji wyniku ogólnego (słowo puste), ponieważ nie ma żadnego słowa kończącego się na jakąkolwiek literę. Następnie należy zoriętować się jak dane słowa(funkcje) zmieniają jedynki i zera w różnych przypadkach. □

Ostateczne rozwiązanie dla każdego podzbioru próbuje dołożyć kolejne słowo, zmieniając pozycję jedynki, więc w naszym dp wystarczy trzymać dp[2^n][27], które będzie równało się dla danego podzbioru słów ile permutacji zawiera 1 na pozycji: ostatecznego wyniku, wyniku z założeniem że a jest ostatnią literą, założeniem że b jest ostatnią literą...