

## به نام خدا

علی غفاری ثابت - ۹۸۲۳۹۰۳

### توضیح کد:

۱- ابتدا تابع calc رو تعریف می کنیم به این صورت که M (سایز هر خط)، sum\_size (آرایه ای که عضو i ام آن برابر حاصل جمع طول کلمه ۰ ام تا کلمه i-1 ام است)، i و j (اندیس های انتها و ابتدای بازه) را دریافت می کند و هزینه قرار دادن بازه داده از کلمات در یک خط را به شیوه توضیح داده شده در جزوه محاسبه می کند. (حاصل جمع طول کلمه z ام تا کلمه i ام برابر  $sum\_size[i] - sum\_size[j]$  است.)

۲- در بدنه اصلی ابتدا m (سایز هر خط) و w (طول هر کلمه) را از کاربر دریافت می کنیم و سپس آرایه sum\_size را اینگونه تشکیل می دهیم که عنصر ۰ ام آن ۰ و عنصر i ام آن از رابطه  $sum\_size[i] = sum\_size[i - 1] + w[i - 1]$  محاسبه می شود.

۳- سپس آرایه p و dp را به طول  $len(w)$  تعریف می کنیم که در ابتدا تمام عناصرش برابر ۰ است. سپس به کمک دو حلقه عنصر i ام اینگونه تعیین می شود که  $dp[i] = \min(calc(m, sum\_size, i, 0), calc(m, sum\_size, i, j) + dp[j - 1])$  به ازای  $1 \leq j \leq i$  و  $p[i]$  برابر j ای است که عنصر i ام dp به ازای آن کمینه می شود. (اگر  $dp[i] = calc(m, sum\_size, i, 0)$  شود  $p[i]$  برابر ۰ می ماند.)

۴- حال تابع print\_line را تعریف می کنیم به این صورت که p و i را دریافت می کند و راه حل را برای اینکه کلمه ۰ ام تا کلمه i ام را باید به چه خط ها بشکنیم تا هزینه کمینه شود را به صورت بازگشتی چاپ می کند.

۵- در نهایت تابع print\_line را برای تمام کلمات ( $print\_line(p, len(w) - 1)$ ) صدا می زنیم و سپس حداقل هزینه ممکن که همان آخرین عنصر آرایه dp است را چاپ می کنیم.

### پیچیدگی زمانی:

برای محاسبه آرایه sum\_size یک حلقه به طول n داریم که پیچیدگی زمانی اش برابر  $\theta(n)$  است و برای پر کردن آرایه های p و dp دو حلقه تو در تو داریم که حلقه خارجی (i) از ۰ تا n-1 زیاد می شود و حلقه درونی (j) از ۰

تا i زیاد می شود پس تعداد کل تکرار برابر است با:  $\sum_{i=0}^{n-1} (i + 1) = n * (n + 1) / 2 = \theta(n^2)$  و تابع calc نیز

پیچیدگی زمانی اش از  $\theta(1)$  است و تابع print\_line چون i فقط می تواند کم شود و شرط اولیه تابع  $i < 0$  است بنابراین این تابع حداکثر می تواند n بار تکرار شود در نتیجه پیچیدگی زمانی آن برابر  $\theta(n)$  است. پس پیچیدگی زمانی کل تابع برابر است با: (در تمام محاسبات بالا n برابر تعداد کلمات است.)

$$\theta(n) + \theta(n^2) + \theta(1) + \theta(n) = \theta(n)$$

### پیچیدگی حافظه:

چون ۳ آرایه اضافه  $p$  و  $dp$  و  $sum\_size$  را به طول های  $n$  و  $n$  و  $n+1$  تعریف کردیم ( $n$  برابر تعداد کلمات است). پیچیدگی زمانی برابر  $\theta(n)$  است.