به نام خدا

گزارش کار پروژه

در این پروژه قصد داریم یک تعداد کلمات با تعداد کاراکترهای مشخص در یک پاراگراف با طول هرخط مشخص که چه تعداد کاراکتر می توانند در هرخط قرار بگیرد بسته بندی می کنیم.بنابراین یک لیستی داریم به طول این کلمات با تعداد کاراکترهای هرکلمه که می خواهیم آن را در این پاراگراف با روش برنامه ریزی پویا(dynamic programming) پیاده سازی کنیم به طوری که هزینه کل خطوط ناشی از شکستن کلمات موجود در لیستی که داریم به تعدادی خط مینیمم شود.در ابتدا ماکزیمم تعداد کاراکترهایی که در هرخط می تواند

قرار بگیرد را به عنوان ورودی می گیریم.یک آرایه را هم در نظر می گیریم که برای نگه داری کلمات با تعداد کاراکترهای هر کلمه است. بنابراین در ابتدا تعداد کلمات را از ورودی می خواهیم و سپس بر اساس تعداد این کلمات در آرایه موردنظر برای هرکلمه تعداد کاراکتر ها را به عنوان ورودی می گیریم ودر آرایه ذخیره می کنیم.

یک تابع wordPacking برای شکستن کلمات برای قرار گرفتن در خط با تعداد کاراکتر مشخص به طوری که هزینه شکستن این لیست کلمات در این پاراگراف به تعدادی خطوطی مینیمم شود. در این تابع که آرایه ی کلمات (تعداد کاراکترهای هرکلمه)و تعداد کلمات و ماکزیمم تعداد کاراکترهایی که در هرخط می توانند قرار بگیرند را به عنوان ورودی می گیرد.

در این تابع یک آرایه دوبعدی برای نگه داری تعداد فضای اضافی بین کلمات در یک خط را در نظر گرفتیم.یک آرایه دوبعدی دیگر برای محاسبه هزینه قرار گرفتن کلمات در هرخط در نظر

میگیریم.با توجه به آرایه extraspaceاگر این مقدار منفی شود مقدار آرایه هزینه بی نهایت می شود که نشان می دهد کلماتی که داریم در یک خط جا نمی شود،اگرextraspace مثبت شود و j=number\_wordشود در این صورت هزینه را برابر صفر قرار می دهیم که نشان می دهد هیچ فضایی به هدر نرفته است. سپس در یک آرایه دیگر با توجه به هزینه های به دست آمده ،با استفاده از روش برنامه ریزی پویا و بااستفاده از این فرمول

,j>0 overalcost[j]=minimum(overalcost[i-1]+linecost[i][i]) 1<=i<=j

یعنی برای محاسبه ی overalcost[4]به مقادیر overalcost[3],overalcost[2], overalcost[1]نیاز داریم و با توجه به فرمول minimmumرا پیدا می کنیم.

در آخر کلمات را در تعدادی خطوط که مینیمم شده است چاپ می کنیم.

**پیچیدگی زمانی :**

با توجه به اینکه در این الگوریتم برای محاسبه کمترین هزینه و محاسبه ی هزینه ی قرار گرفتن کلمات در یک خط و هم چنین محاسبه ی فضای اضافی در هرخط از دو حلقه تشکیل شده اند که به هم وابسته هستندو بستگی به سایز آرایه ورودی دارد بنابراین پیچیدگی زمانی این الگوریتم برابر o(n^2) است.

**نمونه اجرای برنامه:**

