به نام خدا

دانشکده‌ی مهندسی برق و کامپیوتر دانشگاه تهران

طراحی و تحلیل الگوریتم‌ها، نیم‌سال اول سال تحصیلی ۹۷-۹۶

پاسخ تمرین شماره **۲ (برنامه‌نویسی پویا)**

جواب سوال ۱)

dp[s][i] را برابر با تعداد اعداد بین ۱ تا $۲^{۳۲}$ که مجموع ارقام آن ها برابر s بوده و i رقم دارند در نظر می گیریم. بدیهیست i مقادیر بین ۱ تا $\log s$ را تصاحب می کند. جواب نهایی مسئله d[s][0] + d[s][1] + … + d[s][$\log(s)$] می باشد.

مقدار هر dp به صورت زیر به روز می شود.

dp[s][i] = dp[s-0][i-1] + dp[s-1][i-1] + … + dp[s-9][i-1]

توضیح: اگر x سمت راست ترین رقم عددی که مجموع ارقام آن s است و i رقم دارد به عددی i-1 رقمی می رسیم که مجموع ارقامش s-x می باشد.

جواب سوال ۲)

dp[n][k] را برابر با تعداد حالاتی که می توان n را به صورت جمع اعداد کوچک تر از نوشت که مقدار آن ها حداکثر k باشد در نظر می گیریم. جواب نهایی مسئله d[n][n] می باشد. مقدار آن را به صورت زیر به دست می آوریم.

dp[n][k] = dp[n][k-1] + dp[n-k][k]

توضیح: در ساخت عدد n به صورت مجموعه ای از اعداد که مقادیر آن ها حداکثر k باشد، بررسی می کنیم که آیا از k استفاده شده است یا خیر. اگر استفاده نشده باشد یعنی تعداد حالاتی مد نظر است که n را بتوان با اعداد حداکثر k-1 ساخت. اگر لااقل یک بار استفاده شده باشد تعداد حالاتی مدنظر است که n-k را بتوان با اعداد حداکثر k ساخت. زمان محاسبه dp بالا $o(n^2)$ است.

جواب سوال ۳)

dp[n][k] را تقسیم کردن تکه چوب به k قسمت در نظر می گیریم به طوری که در آخرین حرکت خود یکی از قسمت های آن را به k قسمت تقسیم کرده باشیم. جواب نهایی مسئله ما dp[n][2] + d[n][3] + … + d[n][n] می باشد.
مقدار داینامیک به صورت زیر به روز رسانی می شود.

dp[n][k] = dp[n-k+1][2] + dp[n-k+1][3] + … dp[n-k+1][n]

توضیح: وقتی می دانیم در حرکت آخر یک تکه چوب را به k قسمت تقسیم کرده ایم یعنی پیش از آن چوب ما n-k+1 قسمت بوده است. حال جواب به ازای آن را به دست می آوریم.

جواب سوال ۴)

d[i][k] را کمترین طول پوشاندن i نقطه اول با k بازه می نامیم. جواب نهایی مسئله d[n][k] می باشد. این داینامیک به صورت زیر به روزرسانی می شود.

d[i][k] = $\min\big(d[i-1][k-1] + 0. d[i-2][k-1] + w(i-1.i). \dots . d[0][k-1] + w(1.i)\big)$

توضیح: بازه آخر را در نظر می گیریم(بازه ای که پایان آن سمت راست ترین باشد.) فرض می کنیم که این بازه t نقطه را پوشش می دهد(t عددی بین ۱ تا i می باشد.) بنابراین سایر نقاط که i-t نقطه اول می باشند را باید با k-1 بازه پوشاند. منظور از w(i,j) فاصله بین نقاط i و j است.

جواب سوال ۵)

d[i][j] را برابر با جواب مسئله برای رسیدن به سطر i و ستون j می نامیم. جواب نهایی مسئله بیشترین مقدار این داینامیک به ازای همه ی خانه های جدول است.

الف) d[i][j] = a[i][j] + max(d[i-1][j-1], d[i-1][j], d[i+1][j]). توضیح: a[i][j] مقدار هر خانه است و برای رسیدن به آن ۳ راه ممکن است که بین آن ها بیشترین را انتخاب می کنیم.

ب) d[i][j] = $\max(d[i-1][k] + w[i][k][j])$ توضیح: حالت بندی روی ستونی از سطر i انجام می شود که برای رسیدن به خانه i,j وارد آن می شویم و سپس با حرکت عرضی به خانه i,j می رسیم. w[i][k][j] برابر است با میزان امتیاز جمع شده در حرکت عرضی داخل سطر i بین ستون های k تا j که مقدار آن با داینامیک دیگری در همین طول زمانی قابل محاسبه است.

جواب سوال ۶)

d[r][0] را جواب مسئله به ازای زیردرخت راس r تعریف می کنیم. d[r][1] را جواب مسئله به ازای راس r تعریف می کنیم به طوری که در بخش بندی نهایی بخش یک بخش شامل راس r بدون هیچ راس سیاهی باشد.

حال برای محاسبه d[r][0] در نظر می گیریم که اگر r سیاه باشد به ازای هر فرزندش اگر سیاه باشد باید یال بین آن دو را حذف کرد. اگر سفید باشد دو حالت موجود است. یکی این که یال بین آن دو حذف شود و زیردرخت آن راس به d[x][0] حالت تقسیم بندی شود. دوم این که آن یال حذف نشود و زیر درخت آن راس به d[x][1] حالت تقسیم بندی شده که در نهایت آن راس داخل بخشی بدون راس سفید قرارگرفته و با راس r که سیاه هست بخش واحدی را تشکیل دهد. به این صورت d[x][1]+d[x][0] حالت برای آن راس فرزند وجود دارد. حال تمام این d[x][1]+d[x][0] ها و تک حالت های مربوط به فرزندان سیاه را در هم ضرب می کنیم و d[r][0] به دست می آید. d[r][1] طبق تعریف برای راس ریشه سیاه صفر است زیرا هیچ حالتی وجود ندارد که این راس در بخشی بدون راس سیاه قرار بگیرد.

این دو داینامیک را برای راس سفید نیز به این صورت محاسبه می کنیم.
d[r][1] اگر r سفید باشد مانند حالت سیاه بودن r و محاسبه d[r][0] محاسبه می شود. یعنی از بین فرزندانش هر کدام سیاه باشد یال بین آن ها باید قطع شود و تک حالت در نظر گرفته شود. هر کدام که سفید باشند باید d[x][1] آن فرزند در نظر گرفته شود. سپس تمامی این d[x][1] ها و تک حالت ها در هم ضرب شوند.

جواب نهایی مسئله d[r][0] می باشد.