

نمونه امتحان پایان ترم درس طراحی الگوریتم

احمد خادم زاده

<http://khademzadeh.mshdiau.ac.ir/>

دانشگاه آزاد اسلامی مشهد

نیمسال دوم ۱۳۸۸-۱۳۸۷

مرتبه رشد مجانبی - Asymptotic Order of Growth

سوال اول

توابع زیر داده شده اند:

$$\sqrt{n}, 2^n + 30, 100n, 4n^2 - 6n^3 + n^4, n^{lgn}, n^2 + 11n, 2n^2 lgn, (lgn)^2$$

این توابع را به صورتی مرتبی کنید (از چپ به راست بنویسید)، که اگر در لیست شما تابع $f(n)$ در

سمت راست تابع $g(n)$ قرار گرفته باشد، آنگاه داشته باشیم: $f(n) = O(g(n))$

تقسیم و غلبه - Divide and Conquer

سوال اول

برای رابطه بازگشتی زیر، $g(n)$ را مشخص کنید، به نحوی که داشته باشیم $T(n) = O(g(n))$. پاسخ خود را مختصراً تشریح کنید.

$$T(n) = T(n - 3) + 3.n$$

سوال دوم

- یک الگوریتم تقسیم و غلبه بنویسید که با کمک آن بتوانیم بزرگترین عدد موجود در یک آرایه دوبعدی $n \times n$ را بیابیم.
- معادله بازگشتی مربوط به پیچیدگی زمانی الگوریتمی که نوشته اید، را بنویسید.
- با کمک معادله بازگشتی فوق حد بالای مرتبه زمانی این الگوریتم را محاسبه نمایید.

برنامه نویسی پویا - Dynamic Programming

سوال اول

الگوریتم حریصانه خرد کردن سکه فقط برای سکه هایی خاصی (مثل ۲۵، ۱۰، ۵ و ۱) کار می کند. می خواهیم یک الگوریتم پویا برای خرد کردن سکه بنویسیم که برای هر نوع سکه ای کار کند. فرض کنید انواع سکه ای که در اختیار داریم عبارت باشد از $D = \{d_1, d_2, \dots, d_n\}$. رابطه بازگشتی زیر برای این الگوریتم پویا نوشته شده است:

$$c[j] = \begin{cases} 0 & \text{if } j \leq 0 \\ 1 + \min_{1 \leq i \leq k} \{c[j - d_i]\} & \text{if } j > 0 \end{cases}$$

- الگوریتم مربوط به این کار را با کمک رابطه بازگشتی بالا بنویسید.
- پیچیدگی زمانی این الگوریتم را محاسبه نمایید و در مورد نحوه محاسبه خود به اختصار توضیح دهید.

عقبگرد - Backtracking

روش حریصانه - Greedy Method

سوال اول

در فایل ورودی $Input1$ مجموعه حروف $Set1$ با درصد تکرار $Freq1$ را داریم. کد هافمن ($Code1$) را برای عناصر این مجموعه بدست می آوریم. فایل ورودی $Input1$ را با کمک $Code1$ فشرده سازی می کنیم. اندازه فایل فشرده شده حاصل $Size1$ می شود. فرض کنیم دو حرف a و b دارای بیشترین تکرارها در $Input1$ بوده اند و تعداد تکرار هر یک به ترتیب $F1$ و $F2$ بوده است. و فرض کنید که حرف z در فایل $Input1$ وجود نداشته است. در تمام فایل $Input1$ حروف a و b را با حرف z جایگزین می کنیم. این فایل جدید بدست آمده را $Input2$ می نامیم. بدیهی است که در فایل جدید تعداد تکرار حروف a و b و z به ترتیب برابر با 0 و 0 و $F1 + F2$ می باشد. اگر برای $Input2$ هافمن ($Code2$) را مجدداً حساب کنیم و با کمک $Code2$ فایل $Input2$ را فشرده کنیم و اندازه فایل فشرده شده را $Size2$ بنامیم، آنگاه

- آیا $Size2$ از $Size1$ کوچکتر است؟ چرا؟
- آیا برای هر $Input1$ و $Input2$ مقدار $|Size2 - Size1|$ مقدار مشخصی است؟ چرا؟

سوال دوم

درخت هافمن مربوط به یک فایل را تشکیل داده ایم. در درخت حاصل یکی از زیردرختان همه نودها دارای فقط یک نود می باشد. در مورد تعداد تکرار حروف در فایل چه نظری می توان داد؟

گراف - Graph

سوال اول

گرافی دارای n نود و n^2 لبه می باشد. پیچیدگی زمانی الگوریتم پیمایش عمقی برای این گراف از چه مرتبه ای است؟ به اختصار توضیح دهید.

سوال دوم

فرض کنید که گراف $G = (V, E)$ یک گراف متصل بدون جهت باشد و نمایش لیست همسایگی گراف به ما داده شده است. می خواهیم نودی از این گراف را بیابیم که با حذف آن نوع تعداد مولفه های جدا از هم به وجود آمده گراف بیشترین باشد.

- فرض کنید پیمایش عمقی را از گره $r \in V$ شروع می کنیم. چگونه می توانیم $c(r)$ (تعداد مولفه های حاصل پس از حذف نود r) را با کمک این پیمایش عمقی به روش کارایی بدست آوریم؟
- فرض کنید نود $v \in V$ یک نودی غیر از ریشه درخت حاصل از پیمایش عمقی مرحله قبل باشد. همچنین فرض کنید که هیچ یک از فرزندان v لبه برگشتی (backedge) به نودهای بالای v در درخت حاصل ندارند. چگونه می توانیم $c(v)$ را محاسبه کنیم؟