آزمون میان نیم‌سال درس "طراحی و پیاده‌سازی زبان‌های برنامه‌سازی"

تاریخ: ۱۴۰۱/۲/۱۸ زمان آزمون: ۳ ساعت

نکات مهم:

۱- جواب تمام مسائل را به صورت دست نوشته و یا تایپی بنویسید. سپس جواب هر مساله را در یک فایل pdf جداگانه گذاشته و در قسمت مربوطه به آن سوال در کوئرا آپلود کنید. دقت کنید که در کوئرا تنها امکان ارسال pdf وجود دارد. تنها ارسال نهایی شما در کوئرا تصحیح خواهد شد. دقت کنید که در صورت ارسال چند جواب، کوئرا به صورت خودکار آخرین ارسال را به عنوان ارسال نهایی در نظر می گیرد. برای رعایت عدالت و نظم، در زمان برگزاری آزمون به پرسشی پاسخ داده نخواهد شد.

۲- استفاده از هر منبع مکتوب یا اینترنتی در دسترس شخص دانشجو مجاز است. اما مشورت با دیگری (هر کسی که باشد و در هر اندازه) مجاز نیست. لطفا جواب سوال ها را به دیگران منتقل نکنید و از پاسخ دیگران حتی اگر در دسترس شما قرار گرفت استفاده یا کپی نکنید. در صورت وقوع چنین مواردی مطابق با آیین نامه‌های دانشگاه رفتار خواهد شود.

موفق، سلامت و پیروز باشید.

ایزدی

مسائل:

۱. (۴ نمره) اگر یک لیست از اعداد صحیح به نام L و یک عدد صحیح d داده شده باشند از بین تمامی جایگشت‌های ممکن لیست L جایگشت‌هایی که اختلاف هر دو عدد متوالی دلخواه در آنها کمتر یا مساوی d است را *جایگشت‌های هموار* (Permutations Smooth) می‌نامیم. در این مساله و مساله بعد می‌خواهیم به دو روش مختلف تابعی بنویسیم که با گرفتن لیست L و عدد صحیح d لیستی شامل تمامی جایگشت‌های هموار L تولید کند.

i. ابتدا تابع Perms(L) را بنویسید که لیست همه جایگشت‌های ممکن لیست L را تولید کند.

ii. سپس تابع IsSmoothPerm?(L,d) را بنویسید که هموار بودن یا نبودن لیست L بر اساس عدد d را می‌آزماید.

iii. در نهایت با استفاده از روش فیلتر کردن، تابع SmoothPerms(L,d) را بنویسید که لیست تمامی جایگشت‌های هموار L را خروجی می‌دهد.

(دقت کنید که خروجی تابع اول و سوم لیستی از لیست‌ها و خروجی تابع دوم بولین است.)

۲. (۶ نمره) با توجه به هزینه بسیار زیاد اجرای پیاده‌سازی قبل یک روش بهتر برای حل این مساله پیشنهاد می‌کنیم: مجموعه تمامی جایگشت‌های ممکن لیست را در یک ساختار درختی قرار می‌دهیم به طوری که هر مسیر از ریشه به یک برگ، یک جایگشت را مشخص کند. سپس تابعی می‌نویسیم که درخت را به گونه‌ای هرس کند که فقط شامل نمایش جایگشت‌های هموار باشد. در نهایت از روی درخت حاصل شده، لیست تمامی جایگشت‌های هموار را تولید می‌کنیم. با توجه به توضیحات بالا موارد زیر را به ترتیب بنویسید:

i. ساختار درختی برای نمایش جایگشت‌های ممکن یک لیست پیشنهاد کنید و آن را به صورت یک نوع داده‌ای (Data Type) با نام PermTree پیاده‌سازی کنید. (انتخاب ساختار این درخت از بین ساختار درخت چندتایی یا ساختار درخت دودویی، به نظر شما در این که پیاده‌سازی تابعی بقیه مراحل این مساله برای کدام یک ساده تر است بستگی دارد.)

ii. تابعی با نام ListToPermTree(L) بنویسید تا برای یک لیست، درختی که همه جایگشت‌های ممکن آن لیست را در خود دارد تولید کند.

iii. تابعی با نام PermTreeToPerms(T) بنویسید که تمامی جایگشت‌هایی که توسط درخت T نمایش داده شده را به صورت یک لیست تولید کند.

iv. تابع هرس کردن با نام PruneSmooth(T,d) را بنویسید که تمامی شاخه‌های درخت T که (بر اساس عدد d) هموار نیستند را حذف کند و درخت حاصل فقط شامل جایگشتهای هموار باشد.

v. بر اساس این روش جدید مجددا تابع SmoothPerms(L,d) را بنویسید که لیست تمامی جایگشت‌های هموار L را خروجی می‌دهد.

۳. (۳ نمره) الف- برای مجموعه زیر به هر سه روش بالا به پایین، پایین به بالا و قوانین استنتاج تعریف‌های استقرایی را بنویسید.

$$\{(n, f(n), f(n+2)) \mid n \in N, f(0) = 1, f(1) = 2, f(n+2) = f(n+1) + 2f(n)\}$$

ب- در مورد زیر بیان کنید که قوانین مشخص کننده‌ی چه مجموعه‌ای هستند.

- $(1,5) \in S, \ \frac{(n,k)\in S}{(n+2,2k+1)\in S}$

۴. (۴ نمره) الف- در فصل دو کتاب و همچنین در کلاس درس، محیط (environment) و اعمال روی آن را به دو روش مختلف (داده ساختاری یا پروسیجرال) پیاده سازی کرده‌ایم. پیاده سازی پرویسجرال محیط وقتی امکان بزرگ شدن بازگشتی (extend-env-rec) هم به آن اضافه شده است را به طور دقیق بنویسید.

ب- اگر مانند بخش آخر از فصل ۳ کتاب، برنامه به زبان ورودی را ابتدا به یک برنامه بدون نام متغیرها (Nameless) تبدیل کنیم و تابع مفسر یعنی valueof() را هم از پیاده سازی بدون نام استفاده کنیم، در این صورت پیاده سازی پرویسجرال محیط وقتی امکان بزرگ شدن بازگشتی (extend-env-rec) هم داشته باشد را بنویسید.

۵. (۶ نمره) برنامه زیر که به زبان PROC نوشته شده را در نظر بگیرید. خروجی این برنامه را مشخص کنید. تحلیل کنید اگر در خط آخر به جای ۸ عدد دیگری گذاشته می‌شد، خروجی چه بود.

```
let f = proc (maker)
        proc (x)
                if zero? (x)
                then 0
                else if zero? (-(x, 1))
                then 1
    else -(((maker maker) -(x, 1)), -(0, ((maker maker) -(x, 2))))
in let main = proc (x) ((f f) x)
    in (main 8)
```

به کمک این روش و با استفاده از زبان PROC برنامه‌ای بنویسید که بررسی کند آیا یک عدد صحیح مثبت، مساوی توان صحیح بزرگتر از ۲ برای یک عدد صحیح هست یا نه. (یعنی آیا عدد ورودی n به شکل $n = a^b \ , \ a \geq 2 \ , b \geq 2$ هست؟) در صورت مثبت بودن جواب، #t و در غیر این صورت #f برگرداند.

جمع بارم ۲۳ نمره (۳ نمره ارفاقی است.)

موفق باشید