به نام خدا

دستور کار کارگاه مبانی کامپیوتر و برنامه‌نویسی

جلسه نهم

# توسعه برنامه‌های نسبتا پیچیده کار با توابع

۱. رابطه‌ی بازگشتی دنباله‌ی فیبوناچی به شکل زیر می‌باشد:

1 1 2 3 5 …

این رابطه را می‌توان به صورت زیر در زبان C پیاده‌سازی کرد:

#include <stdio.h>

int fibonacci(int n) {

if (n == 1 || n == 2) {

return 1;

}

return fibonacci(n - 1) + fibonacci(n - 2);

}

int main(int argc, const char \*argv[]) {

}

پیاده‌سازی فوق را به گونه‌ای تغییر دهید که عدد n را از کاربر گرفته و جمله‌ی nام فیبوناچی را چاپ کند. اگر n برابر ۱۰۰ باشد برنامه‌ی شما می‌تواند جواب را در چند دقیقه تولید کند؟

همانطور که مشاهده کردید، جواب در چند دقیقه تولید نمی‌گردد، با توجه به اینکه از نظر منطقی برای محاسبه جمله‌ی ۱۰۰ام حداکثر نیاز به محاسبه‌ی ۱۰۰ جمله‌ی قبل می‌باشد چرا این محاسبه به این میزان طولانی شده است؟

برای اینکه دلیل این موضوع را بهتر متوجه شوید برنامه‌ی فوق را به گونه‌ای تغییر دهید که هر بار فراخوانی تابع فیبوناچی را با پارامتر آن نمایش دهد. خروجی‌ها را برای nهای متفاوت بررسی کنید. آنچه می‌بینید با مدرس خود در میان بگذارید. آیا دلیل این طولانی شدن را پیدا کرده‌اید؟ ارتباطی بین این سوال و قسمت امتیازی سوال سوم تمرین پنجم خود می‌بینید؟

۲. تابع فیبوناچی را به صورت غیربازگشتی بنویسید، این بار تابع را با n برابر ۱۰۰ فراخوانی کنید، آیا جواب حاصل در زمان چند دقیقه حاصل می‌گردد؟

اینطور به نظر می‌رسد که راه‌حل‌های بازگشتی در برخی از موارد کارآیی لازم را ندارند، به نظر شما آیا روشی برای بهبود این موضوع وجود دارد یا واقعا استفاده از روش‌های بازگشتی بی‌ثمر است؟

۳. یکی از روش‌های بهبود زمان اجرا در توابع بازگشتی استفاده از روش memorization می‌باشد. در ادامه کد سی تابع فیبوناچی را که با این روش بهبود یافته است می‌بینید:

#include <stdio.h>

// stores fibonacci sequences in the array.

// global variables are initialized to zero in c

int memory[1000];

int fibonacci(int n) {

if (n == 1 || n == 2) {

return 1;

}

if (memory[n] == 0) {

memory[n] = fibonacci(n - 1) + fibonacci(n - 2); // memorizes what it calculated now.

}

return memory[n];

}

به نظر شما این روش چگونه سرعت محاسبه را افزایش می‌دهد؟ آن را برای مدرس خود تشریح کنید.

۴. تابع بازگشتی زیر را یکبار به روش معمول و بار دیگر به روش دیگر memorization پیاده‌سازی کنید.

۵. فرض کنید تابع ریاضی f (همانند آنچه در حال مطالعه آن در درس ریاضی ۱ هستید 😊) به شرح زیر تعریف شده باشد:

و مشتق آن به شرح زیر می‌باشد:

حال می‌خواهیم معادله را به وسیله‌ی کامپیوتر حل کنیم. یکی از روش‌های حل استفاده از روش نیوتون است، روش نیوتون به صورت زیر است: (اگر دوست داشتید در آینده می‌توانید درس محاسبات عددی را به صورت اختیاری اخذ کنید و در مورد این روش‌ها بیشتر مطالعه کنید.)

نیوتون ادعا می‌کند که اگر از یک نقطه‌ی آغاز کنید و از روش وی استفاده کنید به یکی از ریشه‌های این معادله می‌رسید. حال شما روش این مرحوم را با تابع ریاضی که معرفی کردیم و با استفاده از روش بازگشتی پیاده‌سازی کنید. آیا ادعای این مرحوم صحت داشته است؟

*یکی دیگر از روش‌های افزایش سرعت در محاسبات بازگشتی روش برنامه‌نویسی پویا می‌باشد. اگر با این روش آشنا نیستند، در جلسه‌ی آینده که در مورد آرایه‌ها صحبت می‌کنیم با آن آشنا خواهید شد.*