



انستیتوت مطالعات و تحقیقات علمی ایران
تهران



دانشگاه تهران
دانشکده علوم
گروه ریاضی و علوم کامپیوتر

۱۵ / ۱۱ / ۱۳۸۰

بررسی الگوریتم های مسئله هانوی و تعمیم آن

015999

نکارش

مریم پریسا عرفاتی

استاد راهنما:

دکتر هایده اهرابیان

رساله برای دریافت درجه کارشناسی ارشد

در

رشته علوم کامپیوتر

دی ماه ۱۳۸۰

۳۹۱۵۳



جمهوری اسلامی ایران

دانشگاه تهران

دانشکده علوم

بسمه تعالی

اداره کل تحصیلات تکمیلی دانشگاه

احتراماً به اطلاع میرساند که جلسه دفاع از پایان نامه دوره کارشناسی ارشد علوم کامپیوتر خانم مریم پریسا عرفاتی تحت عنوان:

بررسی الگوریتم های مسئله برجهای هانوی و تعمیم آن

در تاریخ ۸۰/۱۰/۲۵ در گروه ریاضی و علوم کامپیوتر دانشکده علوم دانشگاه تهران برگزار گردید. هیأت داوران بر اساس کیفیت پایان نامه، مقالات انتشار یافته، استماع دفاعیه و نحوه پاسخ به سؤالات، پایاننامه ایشان را برای دریافت درجه کارشناسی ارشد در رشته علوم کامپیوتر معادل با ۶ واحد با نمره ۱۹ (با درجه عالی) مورد ارزشیابی قرار داد.

هیأت داوران

سمت	نام و نام خانوادگی	مرتبه دانشگاهی	دانشگاه	امضاء
۱. استاد راهنما	دکتر هایداه اهرایان	استادیار	تهران	
۲. استاد مشاور	دکتر روزبه ترابی	استادیار	تهران	
۳. استاد داور	دکتر سیدمهدی تشکری	استادیار	صنعتی امیرکبیر	

سرپرست تحصیلات تکمیلی دانشکده

رسول اخروی

مدیر گروه

حمید پزشکی

معاون گروه در امور تحصیلات تکمیلی

سیامک یاسمی



دانشکده علوم
آموزش تحصیلات
تکمیلی

چکیده

در این رساله، سریعترین و مشهورترین الگوریتم‌هایی که تاکنون برای حل مسائل برج‌های هانوی^۱ در دو حالت سه برجی و چند برجی مطرح شده است، شرح و مورد بررسی قرار داده می‌شود.

در مورد حل مسئله سه برجی هانوی، الگوریتم بازگشتی لوکاس^۲ [۳] و الگوریتم‌های غیربازگشتی شوتز-لوکاس^۳ [۳ و ۱۱]، میر^۴ [۱]، ار^۵ [۸] و اهرایان-نوذری [۴] مورد بررسی قرار می‌گیرند. برای حل مسئله چند برجی هانوی الگوریتم بازگشتی گوپتا^۶ [۱۰] و الگوریتم غیر بازگشتی لوزو میاو^۷ [۷] مرور گردیده‌اند.

در انتها، برای حل مسائل چند برجی هانوی^۸، یک الگوریتم غیربازگشتی جدید طراحی و ارائه می‌شود که نسبت به الگوریتم‌های مطرح شده در این مورد، از سرعت بالاتری برخوردار می‌باشد.

¹ The Tower of Hanoi

² Lucas

³ Schoutes-Lucas

⁴ B. Meyer

⁵ M.C.Er

⁶ Gupta

⁷ Lu Xue- Miao

⁸ Multipeg Towers of Hanoi

منت فدای را عز و جل که طاعتش موجب قربت است

و به شکراندرش مزید نعمت

تشکر و قدردانی

با تشکر از پدر و مادر مهربانم
که همواره تشویقهایشان
مایهٔ پشتکار و دلگرمی‌ام بوده است.

و با تشکر از همسر عزیزم
که یاری‌دهنده و مشوق من بوده است.

تشکر و قدردانی

مراتب سپاسگزاری عمیق خود را از استاد ارجمند، سرکار خانم دکتر اهرابیان، بخاطر راهنمایی‌های مفید و ارزنده ایشان در طول دوران تحصیل و نگارش این پایان‌نامه ابراز می‌دارم.

از نظرات و پیشنهادات ارزشمند جناب آقای مهندس نوذری کمال تشکر را دارم. همچنین از ریاست متمرک گروه ریاضی و علوم کامپیوتر دانشکده علوم دانشگاه تهران، جناب آقای دکتر رسولیان و اساتید متمرک که دآوری این رساله را برعهده داشتند تقدیر و تشکر می‌نمایم.

فهرست مطالب

صفحه	عنوان
۱	مقدمه
۳	فصل اول: مفاهیم اولیه
۳	۱-۱- الگوریتم
۸	۲-۱- پشته
۱۰	۳-۱- مفاهیم گراف و درخت
۱۴	۴-۱- مسئله برجهای هانوی
۱۷	فصل دوم: مسئله سه برجی هانوی
۱۸	۱-۲- الگوریتم بازگشتی لوکاس برای حل مسئله برجهای هانوی
۲۰	۲-۲- الگوریتم‌های غیربازگشتی برای حل مسئله برجهای هانوی
۴۲	فصل سوم: الگوریتم بازگشتی برای حل مسئله چند برجی هانوی
۵۲	فصل چهارم: الگوریتم غیر بازگشتی برای حل مسئله چند برجی هانوی
۷۷	فصل پنجم: طراحی یک الگوریتم غیر بازگشتی جدید، برای حل مسئله چند برجی هانوی
۹۱	مراجع
۹۳	پیوست

فهرست جداول

صفحه	عنوان
۸	جدول ۱-۱- چند مثال از پیچیدگی زمانی چند تابع
۲۶	جدول ۱-۲- نمونه‌ای از مکان سمت‌راست‌ترین یک در اعداد دودویی
۳۰	جدول ۲-۲- حالات مختلفی که برجهای A، B و C می‌توانند به خود بگیرند.
۳۵	جدول ۲-۳- هر حرکت از مینه مبدأ به مقصد متناظر با یک برجسب روی هر گره از درخت می‌باشد.
۴۸	جدول ۳-۱- محاسبه n_i ها و عمق درخت و تعداد کل حرکات لازم برای حل یک مسئله هانوی.
۵۹	جدول ۴-۱- مراحل اجراء الگوریتم متناظر با مسئله هانوی. برای ۵ برج و ۲۰ دیسک.
۷۱	جدول ۴-۲- محاسبه اعداد f که در یک مسئله هانوی با ۵ برج و $u(3,5)$ دیسک مطرح می‌شوند.
۹۰	جدول ۵-۱- مقایسه زمان اجراء الگوریتم گوپتا و الگوریتم جدید ارائه شده در این رساله. برای حل یک مسئله ۵ برجی هانوی. (تعداد دیسکهایی که با x مشخص شده‌اند، یک درخت پر متناظر با الگوریتم مربوطه، ایجاد می‌کنند).

فهرست اشکال

صفحه	عنوان
۹	شکل ۱-۱- اضافه و حذف کردن عناصر از پشته
۱۰	شکل ۲-۱- نمونه‌ای از نمایش گراف
۱۱	شکل ۳-۱- نمونه‌ای از یک درخت
۱۳	شکل ۴-۱- نمونه‌ای از درختهای دودویی پر و کامل
۱۶	شکل ۵-۱- برجهای هانوی در حالت ۳ برجی
۱۶	شکل ۶-۱- برجهای هانوی در حالت چند برجی
۲۵	شکل ۱-۲- درخت حاصل از فراخوانی‌های بازگشتی در روش بازگشتی حل برجهای هانوی. برای سه برج با ۴ دیسک.
۲۷	شکل ۲-۲- در پیمایش، Inorder، زمانی که قبل از یک گره میانی، یک برگ ملاقات شود.
۳۶	شکل ۳-۲- درخت دودویی متناظر با راه حل برجهای هانوی برای $n=4$.
۴۵	شکل ۱-۳- نمونه‌ای از یک درخت راستگرد $T(3)$
۴۸	شکل ۲-۳- درخت متناظر با مسئله هانوی با ۲۵ دیسک و ۵ برج.
۵۳	شکل ۱-۴- تبدیل یک درخت متناظر با Tower (5,4)- Original به یک درخت دودویی.
۶۸	شکل ۲-۴- درخت متناظر با مسئله هانوی با ۵ برج و $u(5,4)$ دیسک به همراه شماره دیسکهای متناظر با هر گره.
۶۹	شکل ۳-۴- تبدیل یک مسئله هانوی با ۵ برج و ۸ دیسک به یک مسئله هانوی با ۵ برج و ۱۰ دیسک (با افزودن ۲ دیسک مجازی).
۷۸	شکل ۱-۵- درخت حاصل از فراخوانی‌های بازگشتی الگوریتم گوپتا، برای یک مسئله ۵ برجی هانوی با ۱۰ دیسک.

- شکل ۵-۲ - یک درخت چپ گرد، برای حل مسئله هانوی با $p=3$ و $n=10$
- شکل ۵-۳ - یک درخت چپ گرد برای حل مسئله هانوی با $p=3$ و $n=8$
- شکل ۵-۴ - یک درخت چپ گرد، به همراه شماره دیسک‌ها و برج‌ها برای حل مسئله هانوی با $p=3$ و $n=8$.
- شکل ۵-۵ - درخت متناظر با مسئله هانوی با $p=3$ و $n=18$.

مقدمه

در این رساله، به شرح سریعترین الگوریتم‌هایی می‌پردازیم که برای حل مسائل هانوی، در دو حالت سه برجی و چند برجی مطرح شده‌اند. هدف از تألیف این رساله، بررسی الگوریتم‌های موجود و نهایتاً طراحی الگوریتم جدیدی است که مسئله چند برجی هانوی را با سرعت بالاتری نسبت به سایر الگوریتم‌های موجود، حل نماید.

ادوارد لوکاس^۱ ریاضیدان فرانسوی، در سال ۱۸۸۳ میلادی، مسئله سه برجی هانوی و در سال ۱۸۸۹ میلادی نیز مسئله چند برجی هانوی را بعنوان یک بازی ریاضی مطرح نمود.

شرح این مسائل در فصل اول تحت عنوان مفاهیم اولیه، توضیح داده شده است. این رساله شامل پنج فصل و به شرح زیر می‌باشد:

فصل اول - مفاهیم اولیه:

در این فصل مفاهیم اولیه، تعریف می‌شوند. تعاریفی از قبیل الگوریتم، پشته^۲، گراف و درخت [۱۲ و ۱۶ و ۲] که به همراه جزئیات بیشتر مطرح می‌شوند. از آنجا که مطالب فصول بعدی، بر پایه همین مفاهیم بنا شده‌اند، لازم است جهت یادآوری، مروری بر این تعاریف داشته باشیم. در انتها نیز، تاریخچه‌ای مختصر از مسائل هانوی مطرح می‌شود.

فصل دوم - مسئله سه برجی هانوی:

در این فصل به شرح و بررسی الگوریتم بازگشتی لوکاس [۳] و الگوریتم‌های غیربازگشتی شوتز - لوکاس [۱۱ و ۳]، میر [۱]، ار [۸] و اهرابیان-نوذری [۴] پرداخته می‌شود. همچنین پیچیدگی زمان و حافظه در هر یک مورد محاسبه قرار گرفته است.

¹ Edouard Lucas

² Stack

فصل سوم- الگوریتم بازگشتی برای حل مسئله چند برجی هانوی:

فصل سوم مشخصاً به مطالعه الگوریتم بازگشتی گوپتا [۱۰] اختصاص داده شده است.

فصل چهارم- الگوریتم غیر بازگشتی برای حل مسئله چند برجی هانوی:

لوزومیاو [۷] با ارائه یک الگوریتم بازگشتی برای حل مسئله چند برجی هانوی، یک الگوریتم غیر بازگشتی طراحی نمود که در این فصل این دو الگوریتم مورد بررسی قرار داده شده است.

فصل پنجم- طراحی و ارائه یک الگوریتم غیر بازگشتی جدید برای حل مسئله چند برجی هانوی:

فصل آخر به طراحی یک الگوریتم غیر بازگشتی اختصاص یافته که مسئله چند برجی هانوی را با سرعت بالاتری نسبت به سایر الگوریتم‌های ذکر شده، حل می‌نماید.

در پایان به محاسبه پیچیدگی زمان و حافظه و مقایسه سرعت این الگوریتم، با الگوریتم‌های گوپتا و لوزومیاو پرداخته شده است.

امید آنکه این اثر شمع کوچکی در راه پر فروغ محققان و پژوهشگران باشد.

فصل اول

مفاهیم اولیه

در این فصل، مفاهیم اولیه الگوریتم، پشته، گراف و درخت [۱۲ و ۱۶ و ۲] بیان می‌شوند و در ادامه، مسئله برجهای هانوی به همراه شرح مختصری از تاریخچه آن بررسی می‌گردد.

۱-۱- الگوریتم :

الگوریتم مجموعه محدودی از دستور العمل‌هایی است که با دنبال کردن آنها، هدف خاصی برآورده می‌شود. موارد زیر در هر الگوریتم قابل بررسی می‌باشد:

- ورودی: یک الگوریتم می‌تواند یک یا چند ورودی داشته باشد که از محیط خارج تأمین می‌شوند.

- خروجی: الگوریتم باید حداقل، یک کمیت به عنوان خروجی داشته باشد.

- قطعیت: هر دستور العمل باید واضح و خالی از هر نوع ابهامی باشد.

- محدودیت: الگوریتم باید، پس از طی مراحل محدودی خاتمه یابد.

- کارایی: هر دستور العمل باید طوری باشد که با استفاده از قلم و کاغذ بتوان بطور دستی آنرا اجراء نمود. در واقع فقط قطعیت کافی نیست بلکه هر دستور العمل باید انجام پذیر نیز باشد.

رژا انصاری
مدرس

الگوریتم‌های بازگشتی:

یکی از مشخصه‌های قدرتمندی که در ایجاد یک الگوریتم، در نظر گرفته می‌شود، امکان رجوع تابع به خودش، برای حل یک مسئله می‌باشد. این تکنیک کنترلی، بازگشت^۱ نامیده می‌شود و در بسیاری از مسائل که بکارگیری حلقه‌هایی مانند **For** و **While** برای حل مسئله مشکل می‌باشد، بکار برده می‌شود.

در ریاضیات و علوم کامپیوتر، بازگشت به مفهوم رجوع به خود^۲ می‌باشد. بنابراین یک تابع بازگشتی، تابعی است که تعریف آن بر اساس تعریف خودش ایجاد شده است بعنوان مثال، تابع فاکتوریل را می‌توان بصورت بازگشتی تعریف کرد:

$$n! = \begin{cases} 1 & ; \quad n = 0 \\ n \times (n-1)! & ; \quad n > 0 \end{cases}$$

هنگامی که تابع به خودش رجوع می‌کند، از آرگمانهای کوچکتری نسبت به خودش استفاده می‌نماید و مقدار تابع برای حالت یا حالات حداقل، بدون رجوع به خود تعریف می‌شود، بنابراین زنجیر بازگشت به پایان می‌رسد. برای مثال الگوریتم بازگشتی محاسبه فاکتوریل یک عدد را در الگوریتم ۱-۱، ملاحظه می‌نمایید.

```
int Factorial( int n )
{
    if(n<=0) return 1;
    return n*Factorial(n-1);
}
```

لگوریتم ۱-۱: الگوریتم بازگشتی برای محاسبه فاکتوریل

¹ Recursion

² Self- Reference