بسم الله الرحمن الرحيم

مرکز اطلاعات و مدارک علمی ایران

دانشگاه تهران

دانشکده علوم

گروه ریاضی و علوم کامپیوتر

۱۳۸۰/۱۱/۱۵

بررسی الگوریتم های مسئلهٔ هانوی وتعمیم آن

نگارش

مریم پریسا عرفاتی

استاد راهنما:

دکتر هایده اهرابیان

رساله برای دریافت درجهٔ کارشناسی ارشد

در

رشتهٔ علوم کامپیوتر

دی ماه ۱۳۸۰

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 مسئلهٔ 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 ٦ 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 هایده اهرابیان**	استادیار	تهران	اهرابیان
۲. استاد مشاور	**دکتر روزبه ترابی**	استادیار	تهران	
۳. استاد داور	**دکتر سیدمهدی تشکری**	استادیار	صنعتی امیرکبیر	

معاون گروه در امور تحصیلات تکمیلی
سیامک یاسمی

مدیر گروه
حمید پزشک

سرپرست تحصیلات تکمیلی دانشکده
رسول اخروی

دانشکده علوم
آموزش تحصیلات تکمیلی

چکیــده

در این رساله، سریعترین و مشهورترین الگوریتم‌هایی کـه تـاکنون بـرای حـل مسـائل برجـهای هانوی[1] در دو حالت سه برجی و چند برجی. مطرح شـده اسـت، شـرح و مـورد بررسـی قـرار داده می‌شود.

در مـورد حـل مسـئلهٔ سـه برجـی هـانوی، الگوریتـم بـازگشـتی لوکـاس[2][۳] و الگوریتم‌هـای غیربازگشتی شـوتز-لوکاس[3][۱۱و۳]، مـیر[4][۱]، ار[5][۸] و اهرابیـان-نـوذری[٤] مـورد بررسـی قـرار می‌گیرند. برای حل مسئلهٔ چند برجی هانوی الگوریتم بازگشتی گوپتا[6][۱۰] و الگوریتم غیر بازگشتی لوزو میااو[7][۷] مرور گردیده‌اند.

در انتها، برای حل مسـائل چند برجی هانوی[8]، یک الگوریتم غیربازگشتی جدید طراحـی و ارائـه 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 خدای را عز و جل که طاعتش موجب قربت است

و به شکراندرش مزید نعمت

تشکر و قدردانی

با تشکر از **پـــدر و مـــادر مهربانم**

که همواره تشویقهایشان

مایهٔ پشتکار و دلگرمی‌ام بوده است.

و با تشکر از **همســـر عزیزم**

که یاری‌دهنده و مشوق من بوده است.

تشکر و قدردانی

مراتب سپاسگزاری عمیق خود را از استاد ارجمند، سرکار خانم دکتر اعرابیان، بخاطر راهنمایی‌های مفید و ارزندهٔ 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 محترم گروه ریاضی و علوم کامپیوتر دانشکدهٔ علوم دانشگاه تهران، جناب آقای دکتر رسولیان و اساتید محترمی که داوری این رساله را برعهده داشتند تقدیر و تشکر می‌نمایم.

فهرست مطالب

عنوان	صفحه

فهرست جداول

عنـــوان	صفحـه

فهرست اشکال

صفحه	عنـــوان

عنوان	صفحه

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 مسئلهٔ چند برجی هانوی را با سرعت بالاتری نسبت به سایر الگوریتم‌های موجود، حل نماید.

ادوارد لوکاس[1] ریاضیدان فرانسوی، در سال ۱۸۸۳ میلادی، مسئلهٔ سه برجی هانوی و در سال ۱۸۸۹ میلادی نیز مسئلهٔ 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[2]، گراف و درخت[۱۲و٦و۲] که به همراه جزئیات بیشتر مطرح می‌شوند. از آنجا که مطالب فصول بعدی، بر پایهٔ 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- مسئلهٔ سه برجی هانوی:

در این فصل به شرح و بررسی الگوریتم بازگشتی لوکاس[۳] و الگوریتم‌های غیربازگشتی شوتز - لوکاس[۱۱و۳]، میر[۱]، ار[۸] و اهرابیان-نوذری[٤] 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 مسئلهٔ چند برجی هانوی:

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

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

لوزومیااو[۷] با ارائه یک الگوریتم بازگشتی برای حل مسئلهٔ 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 مسئلهٔ چند برجی هانوی:

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

در پایان به محاسبهٔ پیچیدگی زمان و حافظه و مقایسهٔ سرعت این الگوریتم، با الگوریتم‌های گوپتا و لوزو میااو 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 اولیهٔ الگوریتم، پشته، گراف و درخت[۱۲و۶و۲] بیان می‌شوند و در ادامه، مسئلهٔ برجهای هانوی به همراه شرح مختصری از تاریخچهٔ آن بررسی می‌گردد.

۱-۱- الگوریتم :

الگوریتم مجموعهٔ 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 دستورالعمل باید واضح و خالی از هر نوع ابهامی باشد.
- محدودیت: الگوریتم باید، پس از طی مراحل محدودی خاتمه یابد.
- کارآیی: 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.

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[1] 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[2] 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 محاسبهٔ 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 محاسبهٔ فاکتوریل

---

[1] Recursion

[2] Self- Reference