

تاریخ: 1400/03/27

پیوست : ندارد

بسمه تعالی

پاسخنامه آزمون تشریحی الگوریتمهای پیشرفته نیمسال دوم ۱۴۰۰-۱۳۹۹ نام استاد جناب دکتر سید علی رضوی ابراهیمی

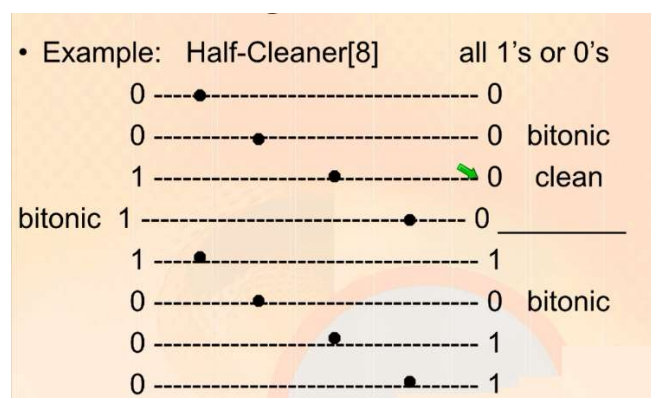
نام دانشجو : سید جواد بسیطی - شماره دانشجویی: ۹۹۰۱۸۹۹۸۹

سوال یک میانترم:

سوال دو میانترم : نحوه عمل نیم پاک کننده در شبکه های مرتب ساز را شرح دهید.
در زمانی که ورودیهای ما بیتونیک (دو آهنگی) ، باشند جهت مرتب سازی آنها، در چندین مرحله عمل میشود. که به هریک از این مراحل نیم پاک کننده اطلاق میشود.
نکته مهم اینکه: هر زمان که توالی دو آهنگی، از صفرها و یکها در ورودی نیم پاک کننده اجرا شود، خروجی حاصل شده بدین صورت است که علاوه بر این که هر دو نیمه بیتونیک هستند و حداقل یک نیمه آنها تمیز شده ، هر المان در نیمه بالایی به اندازه هر المان در نیمه پایینی کوچک خواهد بود .

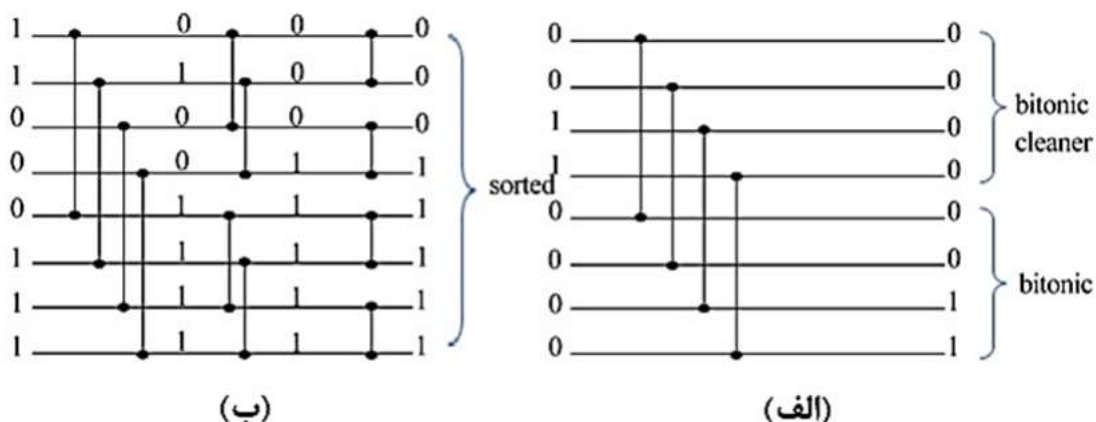
شکل کار بدین صورت است که در شبکه های مقایسه های HALF- CLEANER[n] ورودی برروی خط i با ورودی خط $i+n/2$ مقایسه میشود. فرض بر این است که ورودیها، به صورت بیتونیک یکنواخت افزایشی و یا کاهشی باشند. بعد از پایان مقایسه، حداقل یک نیمه تمیز خواهیم داشت.

مانند شکل زیر که یک توالی دو آهنگی یکنواخت افزایشی را نشان میدهد. همانگونه که مشاهده میشود، بعد از مرحله اول مرتب سازی، در بالا، یک نیم پاک کننده از صفرها و پایین، یک توالی دو آهنگی یکنواخت کاهشی تولید خواهد شد .



نکته: عمق هر نیم پاک کننده، برابر با «یک» بوده و خواهیم داشت: $D(n)=1$

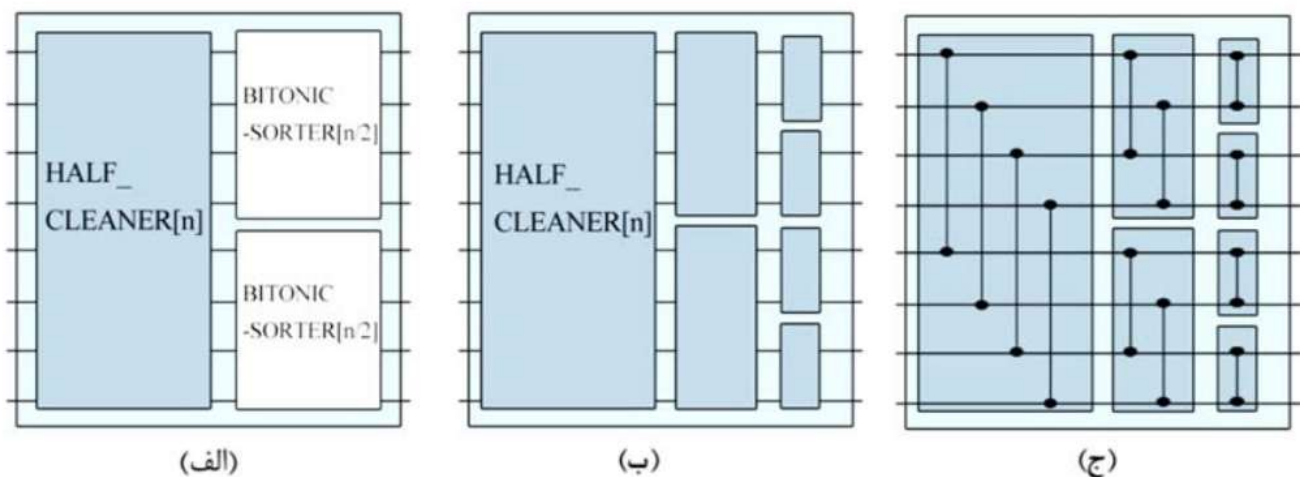
شکل زیر قسمت الف- خروجی توالی شکل گرفته از ورودی صفرها و یک های (00110000) را که به صورت یکنواخت افزایشی است، به روش مرتب سازی دوآهنگی نشان میدهد. همانطور که مشاهده میشود، توالی ورودی، در یک مرحله مرتب شده است. به طوریکه خط i با خط $i+n/2$ مقایسه شده و مقادیر کوچکتر را در نیمه ۲ بالایی، به صورت دوآهنگی پاک قرار میدهد. نیمه پایین نیز دوآهنگی است.



الف) مرتب سازی دوآهنگی برای توالی (۰۰۱۱۰۰۰۰)، (ب) شبکه مرتب ساز دوآهنگی با ۸ ورودی، برای توالی (۱۱۰۰۰۱۱۱)

* ادامه کار به صورت بازگشتی پیش خواهد رفت تا همه نیمه ها مرتب شده شکل گیرند.

به عنوان مثال در شکل بعدی یک شبکه مقایسه BITONIC- SORTER[n] را وقتی که $n=8$ است، برای توالی (11000111) نشان میدهد. این شبکه به صورت بازگشتی و نیم پاک کننده است که هر مرحله مرتب ساز دوآهنگی، روی نیمه از شبکه است. همانگونه که در ساختار بازگشتی شکل مشاهده میشود، یک توالی در نظر BITONIC- SORTER[n] یک به صورت ابتدا، HALF-CLEANER[n] گرفته شده و پس از انجام مقایسه و جابجایی در هر نیمه - به منظور مرتب سازی-، به HALF-CLEANER[n/2] تغییر میکند. به طوریکه، یکی از نیمه ها پاک است و با قرار گرفتن خروجی دارای مقدار کوچکتر در نیمه بالایی و خروجی با برخی مقدار بزرگتر در نیمه پایینی، نیمه ها مرتب میشود و اگر ورودی نیم پاک کننده، دنباله‌ای دو آهنگی از اعداد دلخواه باشد، آنگاه، هر دو نیمه بالایی و پایینی، دو آهنگی بوده و هر عنصر در نیمه بالایی، حداکثر به کوچکی هر عنصر در نیمه پایینی خواهد بود.



سوال یک پایانترم:

سوال دو پایانترم:

سوال سه پایانترم: دسته های سختی مسایل و تفاوت آنها را تشریح نمائید.

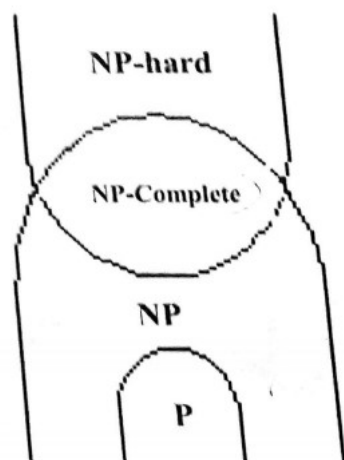
سختی در ذات مسئله نهفته است به طور کلی مسائلی که با الگوریتم های زمان چند جمله ای قابل هستند، حل پذیر یا آسان هستند و مسائلی که به زمان فرا چند جمله ای نیاز دارند حل ناپذیر یا سخت نامیده می شوند. نظریه ای وجود دارد به نام نظریه پیچیدگی محاسباتی که مسائل را بر اساس دشواری آنها دسته بندی می کند این دسته بندی معروفترینشان شامل کلاسهای پیچیدگی زمانی زیر است:

۱- P

۲- NP

۳- NP-Complete

۴- NP-Hard



رابطه بین این کلاسها را تا حدودی به شماتیک روبرو می توان ترسیم کرد:

کلاس P: هر مسئله ای که بتوان در زمان چند جمله ای حل کرد، به کلاس P تعلق دارد. برای این مسائل یک الگوریتم قطعی وجود دارد. و در بدترین حالت به ازای k ثابت در زمان $O(n^k)$ حل می گردد ولی در کلاسهای NP مسایلی که خودش شاید در زمان چند جمله ای حل نشود، ولی اگر یک راه حلش را داشته باشیم، می توانیم درستی آنرا در زمان چند جمله ای واریسی کنیم. یک مسئله در صورتی در کلاس NP-Complete قرار دارد که در NP نیز باشد. طوریکه، هیچ الگوریتم شناخته شده و قابل اجرایی، با زمان چند جمله ای برای آنها وجود نداشته باشد. به عبارت دیگر، پیچیده ترین مسائل کلاس NP در کلاس NPC وجود دارد. به عبارتی مسائل NPC جزء سختترین مسئله ها در کلاس NP هستند. ویژگی این مسائل، استفاده از مسئله های دیگر کلاس NP، برای حل آنها با روش کاهش است. و وجود، یک الگوریتم چند جمله ای شناخته شده برای مسئله های NPC جزء ابهامات فعلی علوم است. در کلاس NP-Hard، کاهش پذیرهای زمان چند جمله ای، ابزاری رسمی برای نمایش سخت بودن یک مسئله با وجود یک عامل زمان چند جمله ای به اندازه مسئله دیگر، فراهم میکند.