

## پروژه نهایی نظریه گراف ها

پارسا فدایی خدمت 97243084

سانیا مسعودی 99243089

خواسته 1:

از روش هایی که برای ساخت گراف هایی با خاصیت OC استفاده کرد این است که با یک گراف کوچک که این خاصیت ها را دارد شروع کنیم و مرحله به مرحله نود و یال اضافه کنیم و حواسمان به این خصوصیات باشد تا در صورت تغییر هر متغیر  $\delta$ ,  $\lambda$  یا  $k$  در مرحله بعدی نود و یا یال را طوری اضافه کنیم که خصوصیات مورد نیاز به گراف برگردد.

خواسته 2:

از نرم افزار هایی که میتوان برای بدست آوردن  $\text{edge connectivity}$  و  $\text{node connectivity}$  استفاده کرد میتوان:

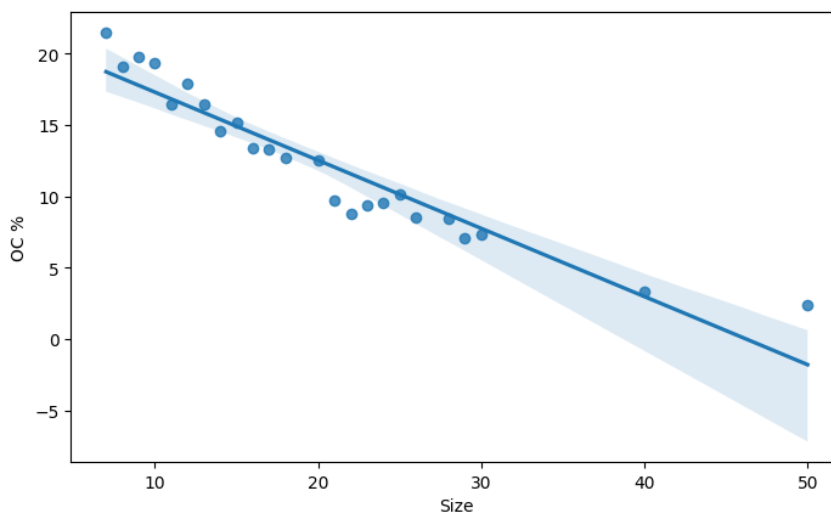
- Network X
- iGraph
- Boost Graph library برای C++
- JUNG: برای جاوا
- Gephi: نرم افزاری جدا برای تحلیل و بررسی و همچنین نمایش گراف ها
- Sagemath

در این نرم افزار ها فقط iGraph و همچنین Sagemath توانایی برای نمایش automorphism دارند و در بقیه باید پیاده سازی شوند.

خواسته 3,5:

تقارن	NS	OC	تعداد یال	اندازه	گراف
Yes	Yes	Yes	6	4	تتراهدرون
Yes	Yes	Yes	12	6	اکتاهدرون
Yes	Yes	Yes	12	8	مکعب
Yes	Yes	Yes	12	8	آکوزاهدرون
No	Yes	Yes	90	60	توپ فوتبال
Yes	Yes	Yes	18	12	تتراهدرون بریده
	قطر	NS			
Yes	4	Yes	32	16	4-هایپرکیوب
Yes	1	Yes	Yes	10	توروس
Yes	5	Yes	Yes	10	رینگ
Yes	2	Yes	Yes	10	Prism-3
Yes	2	Yes	Yes	10	Antiprism
Yes	2	Yes	Yes	10	Twisted Prism
Yes	1	Yes	10	5	$K_{n=5}$
Yes	4	Yes	32	16	$(n,2)$ $n=4$ همینگ

خواسته 4: برای این شبیه سازی ما ابتدا سایز گراف هایی که قرار است شبیه سازی کنیم را مشخص میکنیم و تعداد زیادی (متغیر) از هر سایز تولید میکنیم و درصد فراوانی هر سایز را ذخیره و سپس نمودار میکنیم:



در نموداری که مشاهده میکنیم با افزایش سایز، درصد فراوانی بهینه متصل بودن به 100 میل میکند و از توزیع گیبس پیروی میکند.

خواسته 6 :

گراف	اندازه	تعداد یال	OC	NS	تقارن	L	L <sub>max</sub>	New L	درصد افزایش بار ترافیکی
K <sub>n</sub>	5	10	Yes	Yes	Yes	1	1	1	0%
توروس	5	10	Yes	Yes	Yes	1	1	1	0%
رینگ	5	10	Yes	Yes	Yes	12.5	12.5	21	68%
Prism	5	10	Yes	Yes	Yes	2.3	2.3	2.8	19.9%
Anitprism	5	10	Yes	Yes	Yes	1.5	1.5	1.59	6.66%
Twisted Prism	5	10	Yes	Yes	Yes	1.5	1.5	2	33.33%
همینگ	16	32	Yes	Yes	Yes	3.75	3.75	6.5	73.33%

با خرابی گره ها به صورت تصادفی افزایش بار ترافیکی مشاهده میشود ، در بررسی مشاهده میکنیم که گراف کامل اصلا افزایش بار ترافیکی ندارد اما در گرافی مانند رینگ ما با خرابی مشاهده کردیم که چندین بار گراف جدا میشود، در antiprism بر خلاف پریزم مشاهده کردیم که درصد خرابی بسیار کمتر است و در پریزم چرخیده بعضی اوقات کاهش هم داشتیم. در همه گراف های ما لاندا برابر k بوده اما در صورت کم تر بودن k از لاندا تاثیر خرابی کمتر خواهد بود زیرا مسیر های بیشتری بین گره ها وجود دارد. در صورت وجود گلوگاه یا bottleneck تاثیرات خرابی بسیار روی گراف مشاهده میشود.

رینگ : جدایی

آنتی پریزم: بعضی اوقات برابر

پریزم چرخیده: متفاوت اما همراه با کاهش













