**Теория на игрите**

**от Съзтезателно програмиране**

Направо към: [навигация](file:///D:\Desktop\Copy%20of%20Dani%20С++\Теория_на_игрите.htm#column-one), [търсене](file:///D:\Desktop\Copy%20of%20Dani%20С++\Теория_на_игрите.htm#searchInput)

|  |
| --- |
| **Съдържание**   * [1 Въведение](file:///D:\Desktop\Copy%20of%20Dani%20С++\Теория_на_игрите.htm#.D0.92.D1.8A.D0.B2.D0.B5.D0.B4.D0.B5.D0.BD.D0.B8.D0.B5) * [2 Crazy](file:///D:\Desktop\Copy%20of%20Dani%20С++\Теория_на_игрите.htm#Crazy) * [3 Score](file:///D:\Desktop\Copy%20of%20Dani%20С++\Теория_на_игрите.htm#Score) * [4 Trade](file:///D:\Desktop\Copy%20of%20Dani%20С++\Теория_на_игрите.htm#Trade) * [5 Conqueror](file:///D:\Desktop\Copy%20of%20Dani%20С++\Теория_на_игрите.htm#Conqueror) * [6 Nim](file:///D:\Desktop\Copy%20of%20Dani%20С++\Теория_на_игрите.htm#Nim) * [7 Stripes](file:///D:\Desktop\Copy%20of%20Dani%20С++\Теория_на_игрите.htm#Stripes) * [8 Ресурси](file:///D:\Desktop\Copy%20of%20Dani%20С++\Теория_на_игрите.htm#.D0.A0.D0.B5.D1.81.D1.83.D1.80.D1.81.D0.B8) |

**Въведение**

Всяка комбинаторна игра може да се разгледа като граф, като всеки връх от графа съответства на една ситуация от играта. Ако от една ситуация може да се стигне до друга чрез един ход, то между тези два върха има ребро. Ако от някой връх(ситуация) съществува път(последователност от ходове), след който играчът пак ще се озове в същия връх(ситуация), то играта е циклична. Ако не съществува такъв път, то играта е ациклична.

При всяка смислена игра съществуват крайни позиции, при които играта приключва. Тези позиции се наричат детерминирани, защото е известно кой играч е спечелил или дали играта е равна. Във шаха например детерминирани позиции са, когато единият играч е матиран или настъпи пат.

Всички останали ситуации могат да бъдат определини според детерминираните позиции. Да разгледаме например игра, в която има двама редуващи се играчи и в която са възможни три изхода за всеки играч: победа, равен или загуба. Тогава всеки връх(ситуация) може да се определи, като печеливша(играчът на ход винаги печели ако играе оптимално), губеща(играчът на ход винаги губи при оптимална игра на противника) или неутрална(ако при оптимална игра на двата играча - няма да има победител).

Един връх е печеливш, ако има поне един губещ съсед. Т.е. играча на ход(играч А) може да направи ход и да вкара противника(играч Б) в губеща ситуация(от която противника(играч Б) няма как да спечели при оптимална игра на играч А). По подобен начин, един връх е неутрален, ако не е печеливш и има поне един неутрален съсед. Всички останали не детерминирани върхове са губещи - нямат губещ съсед, нямат равен съсед, следователно имат само печеливши съседи. Т.е. както и да играе играча на ход ще вкара съперникът си в ситуация, от която съперникът му може да спечели при оптимална игра.

От гледна точка на един съзтезател(програмист) или изобщо човек, които цели да напише алгоритъм, който да играе игра с помоща на компютър, всички игри могат да се разделят на два вида - обхватни и необхватни, според броя на ситуациите. Обхватни са тези, които могат да се съхранят в паметта, а необхватни са всички останали(които не могат да се съхранят в паметта).

Ако играта е обхватна и ациклична идеята е просто да се определят всички върхове като печеливши, губещи и така нататък, спазвайки правилата на играта. Ако играта е циклична реализацията е малко по-сложна, но принципът е същият. При необхватните игри нещата се усложняват, защото нещата не могат просто да се вкарат в някакаква рамка(универсален алгоритъм) и понякога трябва да се мисли по-нестандартно.

Ще разгледаме няколко задачи, които могат да се определят като игри и ще разгледаме техните задачи.

**Crazy**

Първата задача е Crazy от пролетния турнир по информатика 2004г. Условието е накратко е следното: Дадени са две естествени числа (a,b) не по-големи от 500. Играят двата играчи, които се редуват. На всеки ход, всеки играч изтрива двете числа и ги замества с (c,b+1)(където c < a и НОД(a,c)>1) или с (a+1,d)(където d < b и НОД(b,d)>1). След това на ход е противникът, като той играе с новите две числа. Играта губи този, който не може да направи ход. Играта трябва да бъде спечелена с минимален брой ходове, като се известно, че противникът играе оптимално, ако му се даде възможност да спечели, ще спечели, ако не може да спечели, ще се играе така, че играта да продължи възможно най-дълго.

Всяка ситуация зависи само от двете числа(лесно се кодира като а\*maxb+b), следователно играта е обхватна. Лесно се доказва, че играта е крайна и ациклична(достъчно е да се покаче, че a+b намалява на всеки ход). В играта няма неутрални ситуации има само печеливши и губещи. Сега лесно чрез едно обхождане в дълбочина можем да определим всички ситуации. Първо обхождаме всичките й наследници(ситуации до които имаме ход) и след това определяме ситуацията, ако имаме поне един губещ наследник, значи ситуацията е печеливша, иначе е губеща.

Допълнително ограничение, че трябва да спечелим играта с минимален брой ходове не е проблем. Можем да намерим броя на ходове оставащи до края при оптимална игра за всяка ситуация. Ако ситуацията е детерминирана, то оставащия брой ходове очевидно е 0. Ако ситуацията е печеливша, то оптималния ход е в губещия наследника, който свършва с най-малко ходове. Ако ситуацията е губеща, то оптималния ход е в наследника, който свършва с най-много ходове.

**Score**

Втората задача е Score от [IOI](file:///D:\mediawiki\index.php\IOI) 2001г. Условието накрато е следното: Даден е граф с N<=1000 върха. Между някой върхове има ребра между други няма(ребрата са еднопосочни). Всеки връх си има собственик(един от двамата играчи), също така във всеки връх има записано естествено число не по-голямо от N(няма повтарящи се числа). Всеки от двамата играчи си има резултат, който в началото е 0. Когато се намираме в даден връх резултатът на собственика на върха става по-голямото число от досегашния му резултат и стойността на върха. След това собственика на върха избира някой от съседните върхове и играта се пренася там. Ако той е собственика на новия връх пак е той на ред, в противен случай на ход е противника му(собственика на върха). В началото играта е във връх 1. Играта свършва, когато играчите отново се върнат във връх едно. Печели този играч, който има по-голям резултат. Както и да играят двамата играчи рано или късно играта ще свърши.

Първо на направим следните наблюдения: тъй като в условието на играта е казано, че тя винаги свършва, то играта очевидно е крайна и ациклична. Също така всички стойности на върховете са различни, т.е. играта не може да завърши на равно. Т.е. всяка ситуация може да бъде определена, като печеливша или губеща. Остава само да определим от какво зависи една ситуация.

На пръв поглед изглежда, че играта зависи от текущия връх и от до сегашния резултат на двамата играчи. За съжаление така ситуациите стават O(N^3), което е твърде много. Затова да се замислим по-задълбочено. Да предположим, че сме в дадена ситуация и досегашния ми резултат е A, а този на противника ми е B.

Нека предположим, че A < B. Това означава, че от всички останали числа(в бъдещето), през който не съм минал, трябва да взема по-голямо число от противника ми. Ако не мога, то очевидно ще загубя. Ако мога да взема число C по-голямо от всички числа, които противникът ми ще вземе в бъдещето, то има два случая. Ако C < B, то тогава отново ще загубя. За да спечеля трябва да взема C > B, т.е. ако мога да вземе по-голямо число от противника ми в оставащите ситуации(в бъдещото), то във всички случай е по-добре това число да бъде максимално.

Нека предположим, че A > B. Тогава ако мога да в оставащите ходове да взема по-голямо число от противника ми, то със сигурност печеля. Ако не мога, има две възможности. Противника ми да вземе число D, което е по-голямо от всички числа, които ще взема по-нататък и D > A, тогава аз ще загубя, а ако D < A тогава ще спечеля. Т.е. ако в оставащите ходове не мога да вземе число по-голямо от противника ми, то винаги е по-добре за мен противника ми да вземе минимално възможно число.

След тези две заключение доказахме, че оптималната стратегия е, ако можем да вземем по-голямо число в бъдещето да вземем максимално такова, а ако не можем да играем така, че противника ни да взема минимално число. Това показва, че оптималната игра не зависи от резултата до момента(от миналото). Въпреки, че в живота не е така бъдещето не зависи от миналото! Следователно една ситуация зависи само от това, в кой връх се намираме, а не с какви резултати сме стигнали. Това прави O(N) = O(1000) ситуации, което е достатъчно малко да се събере в паметта. Следователно играта е обхватна и можем просто да обходим всички ситуации в дълбочина и да определим оптималния ход от всяка от тях в зависимост от наследниците й.

**Trade**

Следващата игра е Trade от пролетния турнир по информатика 2003г. Условието накратко е: Дадени са P сделки, N различни стоки всяка в собствен склад. В склад i в началото има Ki единици стока, и може да се побере максимум Si единици стока. Всяка сделка е от вида - C единици от стока A се разменят за D единици от стока B. Една сделка може да бъде извършена ако в склад за стока B има поне D единици стока и в склад за стока A има място за поне още C единици. Известно е, че Si>=1 и S1\*S2\*S3\*...\*SN\*P <= 1 000 000. Играят двама играча като се редуват да правят сделки. Този, който не може да направи нито една сделка, губи играта. Програмата на съзтезателя трябва да избере дали да играе първа или втора, като ако има начин да спечели трябва да го направи, а ако не трябва да играе до безкрай.

Всяка ситуация зависи само от количествето на стоките във всички складове. Поради ограничението, че произведението на максималните количества стока в складовете е не по-голямо от 1 000 000, играта е обхватна и всяка ситуация може да се кодира. Лесно се забелязва, че играта е циклична, т.е. от някои ситуации съществува последователност от сделки, след които отново се побада в същата ситуация. Има три възможни изхода от играта за всеки играч: победа, загуба или безкрайна игра. Следователно всяка ситуация е или печеливша, или губеща, или неутрална(при оптимална игра от двамата играчи нито единия не може да спечели и се играе безкрайно).

За да решим задачата прилагаме обхождане в ширина като разглеждаме играта отзад напред, т.е. от детерминирани ситуации(края на играта). Детерминираните ситуации са всички, от които не може да се направи нито един ход - те са определени като губещи(добавяме ги в опашката). Като изкарваме една ситуация A от опашката гледаме всички нейни предшественици B(ситуациите от които може да се стигне до дадената чрез една сделка). Ако B е определена продължаваме нататък. Ако B не е определена и A е губеща, то определяме B като печеливша(и я вкарваме в опашката). Ако B не е определена и A е печеливша, отбелязваме, че маркираме наследник A на B. Ако връх B няма други немаркирани наследници, то определяме връх B като губещ(и го вкарваме в опаката).

След края на обхождането в ширина имаме следното: определили сме всички върхове, които имат губещ наследник като печеливши(имаме печеливш ход), определили сме всички върхове, които нямат или имат само печеливши наследници като губещи(каквото и да правим губим при оптимална игра на противника), всички останали върхове(непосетени) са неутрални, защото имат наследници, нямат губещ наследник и не всички наследници им са печеливши. Следователно всички неутрални върхове имат поне един неутрален наследник, т.е. от тях при оптимална игра се играе до безкрай(ако някой играч не играе оптимално ще загуби).

**Conqueror**

Следващата задача е Conqueror от [CEOI](file:///D:\mediawiki\index.php\CEOI) 2002г. Правилата са следните: "Завоевателят" има до 1 000 000 000 войника разположени на N<=2 000 стълби. На всеки ход "Завоевателят" разделя войниците на две групи, след което "Командирът" избира коя от двете групи да напусне играта и коя да остане. След това всички войници от останалата група се качват по едно стъпало нагоре. Ако до най-горното стъпало стигне поне един войник, то играта печели "Завоевателят" - замъка е завладян, в противен случай печели "Командирът". Програмата на съзтезателя трябва да играе "Завоеватела"(хубава роля), като ако има шанс да спечели играта - да спечели, в противен случай - да загуби коректно(без да нарушава правилата на играта).

Ако на всеки ход махаме поне един войник, то войниците все някога ще свършат, а ако не махаме то все някога ще стигнат до горе. Следователно играта е крайна и ациклична. От далече си личи, че ситуациите в тази игра са страшно много и няма как да бъдат побрани в паметта. Следователно играта е необхвата и трябва да оценим дали една ситуация е печеливша или губеща без да изчисляваме всички достижими ситуации. Нека да разгледаме как да определим една игра като печеливша. Една ситуация е печеливша ако можем да закараме 1 войник на най-горното стъпало(номер 1). Един войник на стъпало 1 е равен на двама войника на стъпало 2(защото можем да разделим двата войника в двете групи и независимо от избора на "Командира" да качим един войник до стъпало 1). По същата логика един войник на стъпало 2 е равен на двама войника на стъпало 3 и т.н.т.

Сега като знаем коя ситуация е печеливша достатъчно е да намерим печеливш ход от нея. За да сме сигурни, че можем да спечелим играта и в следващия ход трябва да разделим войниците на две части, така че и в двете стойността на всички войници да е поне 1/2 от войник на стъпало 1. По този начин на следващия ход стойността на оцелелите войници ще се умножи по две и отново ще сме в печеливша ситуация независимо от избора на "Командира".

Нека докажем, че от всяка печеливша ситуация можем да изберем група войници със стойност точно 1/2 войник на стъпало 1. Започваме от стъпало 2 ако имаме поне един войник го взимаме и приключваме, в противен случай търсим 2 войника на стъпало 3. Ако имам достатъчно ги взимаме и приключваме, ако не взимаме толкова колкото можем, умножаваме остатъка по 2 и продължаваме на следващото стъпало. Забележете, че никога няма да надвишим нужния броя войници, а рано или късно войниците ще стигнат, защото ситуацията е печеливша. Така всяка печеливша група от войници я разделяме на две части, в едната част има точно 1/2 войник на стъпало 1, а в другата остават поне толкова.

**Nim**

Една много известна необхватна игра е играта на Ним. Правилата са следните: Дадени са N купчинки с различен брой топчета. Редуват се двама играчи като на всеки ход, играча трябва си избере една купчинка и от нея да махне определен брой топчета. Ако играча иска може да махне цялата купчинка, но е длъжен да махне поне по едно топче. Играта губи този играч, който не може да направи ход.

Очевидно играта е необхватна при по-големи ограничения, защото броят на възможните ситуации е равен на произведението на големината на всички купчинки. Тъйкато сумата от големините на всички купчинки намалява при всеки ход, то следва, че играта е крайна и ациклична.

Тъй като играта е необхватна трябва да определим ситуациите, без да определяме всички наследници до детерминираните ситуации, т.е. трябва да разсъждаваме по друг начин. Единствената детерминирана ситуация(при която играта завършва), е когато всички купчинки са празни, тази ситуация е губеща.

Трябва да определим критерии, според които да различаваме печелившите от губещите ситуации, критерии, който да ги разделя на две множества, така че от първото множеството(съдържащо крайната ситуация) да може да се достига само до елементи от второто множество, а от ситуация във второто множество да може да се достигне до поне една ситуация от първото множество. Първото множество очевидно е множеството на губещите, второто е множеството на печелившите.

Обаче далеч не очевидно е, че този критерии е SG = [xor](file:///D:\mediawiki\index.php\Xor)-а на размерите на всички купчинки. Ако той е равен на 0, то ситуацията е губеща, иначе ситуацията е печеливша. Сега да докажем, че този критерии изпълнява нужните свойства.

Първо, крайната ситуация(всички купчинки са равни на 0) има SG = 0, т.е. ситуацията е губеща. Второ, като сме в дадена ситуация с SG = 0, то като вземем топчета от някоя купчинка, ще променим размера на тази купчинка и следователно ще променим и SG и той ще е различен от 0, т.е. ще стигаме до елементи само от другото множество(само до печеливши). Трето, ако SG не е нула, то разглеждаме най-старшия му значещ бит(равен на 1) в двоичния му запис, нека този бит е бит X. След като бит X е 1 следователно съществува нечен брой купчинки, които имат 1 в същия бит X в двоичния си запис, т.е. съществува поне една такава, нека нейният размер е A. Лесно се забелязва, че SG [xor](file:///D:\mediawiki\index.php\Xor) A < A(очевидно в SG [xor](file:///D:\mediawiki\index.php\Xor) A бит X е равен на 0, а всички по-големи битове са както в A). Т.е. можем да отнемем от A толкова топчета, че купчината да стане равна на SG [xor](file:///D:\mediawiki\index.php\Xor) A. Така ще получим нова ситуация с [xor](file:///D:\mediawiki\index.php\Xor) на всички купчинки равен на SG [xor](file:///D:\mediawiki\index.php\Xor) SG = 0. Следователно имаме поне един наследник от другото множество.

**Stripes**

Последната задача е Stripes от първи кръг на Полската олимпида по информатика 1999/2000г. Условието на играта е следното: Дадена е дъска с размер 1xN (N<=1000) разграфена на N квадратчета. В началото дъската е празна. Има три вида фигури с размери 1xA, 1xB и 1xC, като от всеки вид фигура има безброй много. Редуват се двама играчи, като последователно правят ходове. Един ход се състой в слагане на един от трите вида фигури върху дъската, така че да не се припокрива с вече поставени фигури. Губи този играч, който не може да направи ход(да постави фигура). По зададени до 1000 размери на дъски да се определи дали за играча на ход съществува печеливша стратегия(не зависимо от ходовете на противника) или не.

Играта очевидно е крайна, ациклицна и необхватна(защото достижимите ситуации са твърде много). Нека разгледаме празна дъска с големина 1xK. Като поставим фигура с размер 1xA, разделяме дъската на две независими(фигурата ги разделя). Т.е. вече все едно играем две игри от същия тип, като можем да караме или в едната или в другата игра. Т.е. вече играем сума от две игри.

Ако и двете нови игри са губещи за себе си, очевидно ситуацията е губеща, ако едната ситуация е губеща, а другата печеливша, то ситуацията е печеливша, защото можем да караме ход и да направим и двете нови игри губещи и противникът ни да е на ход. Проблема идва, когато и двете игри са печеливши. Като погледнем играта Ним с две непразни купчинки, то очевидно двете купчинки са печеливши ако се разглеждат като самостоятелни игри. Тогава ситуацията може да губеща или печеливша и няма как да бъде определена само от съседите си само с понятията губещи и печеливши.

За да събираме игри трябва да разширим(обобщим) понятието "печеливша ситуация". За целта използваме SG-числото на играта, което идва от Sprague-Grundy(имената на двамата създатели на метода). SG числото на една ситуация се дефинира, като най-малкото SG до което не може да се достигне от наследниците на ситуацията. Например, ако дадена ситуация може да достигне до ситуации с SG 0, 1, 2, 2, 4, 6, то нейното SG е равно на 3.

Сега можем да разгледаме играта Ним по друг начин - като сума от няколко игри(няколкото купчинки), като SG числото на всяка купчинка поотделно е просто размера на купчинката, защото от нея може да се направи ход във всички ситуаци с по-малко SG, но не може да се направи ход в ситуация със същото SG. Как може би вече се досещате, ако сме в ситуация състояща се от N независими игри с SG стойности a1, a2, a3, ..., aN, то SG стойността на ситуацията е a1 [xor](file:///D:\mediawiki\index.php\Xor) a2 [xor](file:///D:\mediawiki\index.php\Xor) a3 ... [xor](file:///D:\mediawiki\index.php\Xor) aN. Доказателството на този факт оставям на читателят, но то е абсолютно същото, както и в играта Ним.

Сега да се върнем на Stripes. Нека си представим, че поставяме фигура на празна дъска с размер A, като я разделим на две части B и C. Това означава, че от дъска с размер A можем да стигнем до две независими игри. Т.е. SG на ситуацията, която можем да достигнем е SG(B) [xor](file:///D:\mediawiki\index.php\Xor) SG(C). Така като видим всички възможни ходове от дъска с размер A, намираме най-малкото SG, което не можем да достигнем. По този начин можем да определим всички ситуации в играта. Ако SG на една ситуация е равно на 0, то тази ситуация е губеща, а ако не е 0, то значи можем да я направим 0 с един ход, т.е. ситуацията е печеливша.

**Ресурси**

* Тренировки на националния отбор 2003-2005г. с огромната подкрепа на Велин Цанов.
* <http://www.math.ucla.edu/~tom/Game_Theory/comb.pdf>