

Opgave 1 Inbreken in de gevangenis

Concurrency 2011/2012

Achtergrond

Superschurk Evil Genius heeft geheim agent X weer gevangen genomen, en opgesloten in zijn zwaar bewaakte gevangenis. Om de gevangenis binnen te komen moet een 9 cijferige code ingevoerd worden. Het hoofdkwartier weet dat Evil Genius een oud, buitenlands bankrekeningnummer hiervoor gebruikt. Aangezien alleen bekend is in welke range het bankrekening nummer valt, wil agente Y graag weten hoeveel mogelijke bankrekeningnummers er in die range vallen.

Opdracht

Schrijf een programma dat in een bepaalde range kijkt hoeveel van die getallen een bankrekeningnummer kunnen zijn. Dit kan u testen door middel van een aangepaste elfproef. Bij de elfproef worden alle cijfers gewogen bij elkaar opgeteld, afhankelijk van hun positie, om vervolgens te controleren of deze som een elf-voud is. Het eerste cijfer wordt met 9 vermenigvuldigd, het tweede cijfer met 8, etc.

*Een voorbeeld met bankrekeningnummer 27.48.56.190: $2*9+7*8+4*7+8*6+5*5+6*4+1*3+9*2+0*1=220 == 20*11$*

Echter, er is alleen bekend dat het getal een m -voud moet zijn. Dit hoeft dus geen 11 te zijn.

Het programma moet met meerdere threads tegelijkertijd kunnen zoeken. De threads moeten met een gedeelde counter bijhouden hoeveel mogelijke nummers ze hebben gevonden. Threads mogen niet lokaal een counter bijhouden. De gehele range moet eerlijk verdeeld worden over de verschillende threads, zonder gaten of overlap.

Tevens moet u zelf een TaS-lock, een TTaS-lock en een Array-lock implementeren, de drie verschillende versies vergelijken op basis van rekestijd. Schrijf uw bevindingen in maximaal 30 regels in het commentaar bovenaan uw programma. Geef hierbij tevens een korte analyse over welke situaties welke locks wenselijk zijn.

Invoer

De invoer bestaat uit 5 positieve getallen. Het getal op de eerste regel noemen we l , en deze bepaalt welk lock-type het programma gaat gebruiken. Bij een 0 dient er gebruik gemaakt te worden van een TaS-Lock, bij een 1 van een TTaS-lock, en bij een 2 van een Array-lock. Niet voldoen aan deze specificatie verhoogt de kans dat DOMJudge u een “false-positive” zal geven.

Het getal op de tweede regel noemen we b , en geeft de inclusieve ondergrens van het bereik aan waarbij geldt $0 \leq b < 1000000000$ (10^9). Het getal op de derde regel noemen we e , en geeft de exclusieve bovengrens aan waarbij geldt $b < e \leq 1000000000$ (10^9). Het getal op de vierde regel noemen we m , en geeft de modulo aan, waarbij $0 < m \leq 256$. Het getal op de vijfde regel noemen we p . Deze geeft aan hoeveel threads tegelijkertijd zullen gaan zoeken. Voor p geldt dat $1 \leq p < 256$.

Uitvoer

De uitvoer bestaat uit slechts één getal: het aantal getallen in de range $[b,e)$ dat aan de m -proef voldoet.

Voorbeeld

Stel de invoer is:

```
0
274856170
274856190
11
4
```

Dan is de uitvoer 2. (274856174 en 274856182)

Bonus

Achtergrond

Agente Y heeft de password database gekraakt en de SHA1-hash te pakken gekregen. Deze hash is hexadecimaal weergegeven zoals ook te zien op <http://jssha.sourceforge.net/>. Zorg dat ze zo snel mogelijk naar binnen kan.

Opdracht

Pas het programma zo aan dat op de zesde regel aangegeven wordt of de SHA-1 hash bekend is, zie ook Invoer hieronder. U dient de bankrekeningnummers als string te hashen. Indien de hash bekend is, zorg dan dat alle threads stoppen met zoeken zodra de juiste code gevonden. .NET heeft de mogelijkheid om strings met SHA1 te hashen in System.Security.Cryptography. Let er op dat uw programma nog steeds bij elke juiste invoer moet kunnen termineren.

Invoer

De eerste 5 regels blijven gelijk. Het getal op de zesde regel noemen we h , en geeft aan of de hash bekend is of niet. Bij 0 is de hash niet bekend, en is het onbelangrijk wat u op de zevende regel invoert, bij 1 is de hash wel bekend.

Op de zevende regel komt de SHA1-hash te staan die Agente Y gestolen heeft. Dit is een 40-tekens ASCII code. Als u invoer wilt testen die niet als voorbeeld gegeven is dient u deze hash zelf op te zoeken.

Uitvoer

Indien $h == 0$ dan is de uitvoer gelijk aan de standaardopgave. Indien $h == 1$ dan bestaat de uitvoer uit één 9 cijferig getal: de toegangscode. Mocht er geen getal voorkomen wiens hash gelijk is aan de gegeven hash dient de uitvoer '0' te zijn.

Voorbeeld

Stel de invoer is:

```
0
274856170
274856190
11
4
1
c736ca9048d0967a27ec3833832f7ffb571ebd2f
```

Dan is de uitvoer 274856182

Stel de invoer is:

```
0
274856170
274856190
11
4
0
2      (onbelangrijk)
```

Dan is de uitvoer 2.

Over de Opdracht

U dient deze opdracht in tweetallen te programmeren in C# of Java. Hierbij dient u zelf de verschillende locks uit te programmeren. Als u een tweetal gevormd heeft, of als u zelf geen partner kan vinden, kunt u dit aangeven bij de practica, of mailen naar S.E.vanderHurk1@students.uu.nl.

Inleveren

Bij het inleveren van opgaven voor het practicum van Concurrency 2011/2012 wordt gebruik gemaakt van het Domjudge systeem.

Dit systeem controleert automatisch of het programma dat u inlevert:

- correct compileert
- op een aantal (geheime) testinvoeren correcte uitvoer genereert
- op deze testinvoeren snel genoeg werkt.

Indien uw programma aan minstens één van deze voorwaarden niet voldoet, dan wordt uw programma niet geaccepteerd. U krijgt een melding waarom het programma niet wordt geaccepteerd. Daarna heeft u de mogelijkheid om uw programma te verbeteren. U kunt het programma zo vaak inleveren als u wilt, uiterlijk tot de deadline van deze opdracht (zie hieronder).

Zolang het systeem niet een programma voor deze opdracht van u heeft geaccepteerd, is de opdracht nog niet gehaald. Wanneer het systeem wel een programma van u voor deze opdracht heeft ingeleverd, dan is de opdracht meestal gehaald (zie hieronder).

Om de opgave in te leveren gaat u naar

<https://concurrency.cs.uu.nl/team>,

logt u met uw solisID in, kiest u boven in het kopje “Submit” en upload u uw bestand en selecteert u het juiste “Problem” en “Language” in de uitklapmenu’s. Selecteer ook of u de standaard versie, of de versie inclusief de bonus inlevert. Het resultaat van deze submitte kunt u terugvinden onder het kopje “Submissions”.

Aangezien u in tweetallen werkt hoeft u de opdracht alleen in te leveren op het account met het laagste studentnummer. Vergeet niet om bovenaan de opdracht duidelijk beide namen en studentnummers te zetten.

Start Inleveren

Inleveren kan vanaf donderdag 15 september.

Deadline

U moet uw opgave inleveren uiterlijk woensdag 29 september 23.59 uur via het systeem zoals boven beschreven.

Becijfering

Voor het standaard gedeelte kunt u maximaal een 8 behalen. Met het bonus gedeelte erbij is er maximaal een 10 te behalen. Let er op dat de uitleg over de verschillende locks, alsmede passend commentaar bij uw code waaruit u laat blijken dat u snapt waarom u bepaalde keuzes maakt, deel zijn van het cijfer. Dit cijfer telt voor 10% mee van uw eindcijfer.

Fraude en plagiaat

U dient de opdrachten in tweetallen te maken. Het overnemen van code van andere studenten of het kopiëren van code van andere bronnen (zoals het internet) is ten strengste verboden, met als enige uitzondering de code voor de SHA-1-hashing van de strings. Wanneer dit soort gevallen van fraude en/of plagiaat worden ontdekt, dan wordt de procedure, omschreven in het examenreglement van de studie gevolgd. Dit zal in het algemeen leiden tot verdere uitsluiting van deelname aan dit vak dit jaar en melding aan de examencommissie, maar kan ook verdere uitsluiting van de studie als gevolg hebben.

Wedstrijd

Bij deze opdracht is het de bedoeling dat uw programma zo snel mogelijk onze testinvoer verwerkt. Daarom stellen wij twee chocoladerepen beschikbaar: één voor de snelste standaardopgave, en één voor de snelste bonusopgave. Winnaars zullen tijdens college bekend gemaakt worden.

Waarschuwing

Sommige testdata kan er voor zorgen dat uw programma langer aan het rekenen is dan dat u wellicht van Datastructuren gewend bent (denk in de orde minuten). Dit betekent dat u soms even zal moeten wachten voordat DOMJudge u respons geeft. Indien u er voor kiest om uw opdracht pas om “1 voor 12” in te leveren, is de kans daarom groot dat u niet op tijd feedback van DOMJudge zal krijgen. Begin dus op tijd.

Vragen

Vragen over de opdracht kunt u stellen tijdens de werkcolleges van Concurrency. Vragen over het inleveren via het Domjudge systeem kunt u stellen aan Sander van der Hurk (S.E.vanderHurk1@students.uu.nl).