Universität Stuttgart
Fakultät 5, Institut FMI
EfidI 1, Professor Claus

WS 2002/2003
Bertele, Lewandowski
24. Januar 2003

Aufgabenblatt 13

Abgabe bis 31.1.2003, 20:00 Uhr – Besprechung in der Woche ab dem 3.2.2003

Aufgabe 1 Risiko-Lebensversicherung (Zusatzaufgabe, mittel) 4 Punkte

Das war der zweite Teil der Klausur vom Herbst 1999. Diesen Teil mussten nur diejenigen bearbeiten, die eine zweistündige Prüfung über die EfidI1 schreiben mussten. Für diesen Teil gab es 30 Punkte und 60 Minuten Zeit. Hier gibt es jedoch nur die oben angegebenen Punkte. Die Punkte innerhalb des Texts dienen der eigenen Information.

Bei der Risiko-Lebensversicherung muss der Kunde einen monatlich gleichbleibenden Beitrag b während der Vertragslaufzeit l an die Versicherung zahlen. Falls der Kunde während der Vertragslaufzeit stirbt, zahlt die Versicherung den Angehörigen die Versicherungssumme x aus. Falls der Kunde während der Vertragslaufzeit nicht stirbt, darf die Versicherung die gezahlten Beiträge behalten, und der Kunde erhält nichts davon zurück.
Sie sind Mathematiker bei der Banania Lebensversicherungs AG und sollen die schon seit Jahren anhaltend gute Ertragslage des Unternehmens noch weiter verbessern. Da im Vertrieb keine größeren Reserven mehr stecken, kommt der Vorstand auf die Idee, eine neue Tarifstruktur einzuführen, um mehr Gewinn aus den Kunden herauszuholen.
Da das Sterberisiko mit zunehmendem Alter der Versicherten steigt, kommen Sie auf die Idee, statt des bisher festen Beitrags einen neuen variablen Beitrag einzuführen, der mit zunehmender Laufzeit immer höher wird. Der neue Beitrag soll daher jeden Monat um einen festen Promillesatz s steigen.

Wenn Sie es schaffen, ein Ada95 Programm zu schreiben, mit dem sich die Gewinnerwartung berechnen lässt, freuen sich mehrere Leute: Die Versicherten zahlen am Anfang einen viel niedrigeren Tarif (sie rechnen aber nicht aus, wie hoch der Beitrag im Laufe der Zeit steigt, denn dann würden sie sich nicht mehr freuen, sondern nur noch ärgern), der Vorstand reibt sich die Hände und Sie werden zum Chefmathematiker befördert. Die Gewinnerwartung ist ein sogenannter Erwartungswert, also ein Wert, der im Durchschnitt erwartet werden kann, wenn sehr viele Kunden einen Vertrag bei der Banania Lebensversicherungs-AG abschließen. Die Berechnung des Erwartungswerts ist weiter unten beschrieben.

Ihr Programm muss am Anfang folgende Daten interaktiv (mit *kurzer* Texterklärung) abfragen:
Versicherungssumme x (DM als `Float`-Zahl), Eintrittsalter e (ganze Monate), Laufzeit l (Monate), Anfangsbeitrag a (DM), Steigerungsrate des Beitrags s (Promille).

Bereits vordefiniert ist eine Tabelle der Lebenserwartungen, die folgendermaßen aussieht:

```
package Daten is
Stirb: Array(1..1200) of Float;
end Daten;
```

Dabei gibt `Stirb(m)` die Wahrscheinlichkeit an, dass ein Mensch im Lebensmonat *m* stirbt. Es gilt $\sum_{m=1}^{1200} Stirb(m) = 1{,}0$.
Sie können davon ausgehen, dass die Werte in dieses Array bereits eingetragen sind.
(24 Punkte für Näherungslösung): Ihr Programm soll für jeden Monat der Laufzeit tabellarisch ausgeben: Lebensmonat *m*, Monatsbeitrag *b*, Monatlicher Erwartungswert des Risikos *r*, Erwarteter Gesamtgewinn *g*. (*r* und *g* sind Durchschnittswerte für eine sehr große Versichertenanzahl, die alle die gleiche Versicherung mit denselben Daten abschließen wollen.)
Die Berechnung der Näherungslösung geschieht folgendermaßen:
Zunächst wird der am Anfang des Monats zu zahlende Beitrag *b* berechnet.
Sodann wird näherungsweise der Erwartungswert des Risikos *r* durch Multiplikation von `Stirb(m)` mit *x* berechnet. Der erwartete Gesamtgewinn *g* erhöht sich um *b* und wird um *r* vermindert.
(6 Punkte für genauere Lösung): Berücksichtigen Sie zusätzlich, dass nach Fälligkeit der Versicherung (wenn der Versicherte stirbt) kein Beitrag weiterbezahlt wird. Dazu können Sie die Wahrscheinlichkeit *w* berechnen, dass in einem Monat *m* überhaupt ein Beitrag bezahlt wird. Im Sterbemonat selbst wird der Beitrag zum letzten Mal bezahlt, danach nicht mehr.
(bis 4 Zusatzpunkte) Wer zusätzlich berücksichtigt, dass nicht alle Personen das Eintrittsalter *e* überhaupt erreichen (Korrektur der internen Abhängigkeit von `Stirb`, da $\sum_{i=e}^{1200} Stirb(i) < 1{,}0$), erhält Sonderpunkte und wird zum Oberchefmathematiker befördert!

Aufgabe 2 Fibonacci-Zahlen I (Votieraufgabe, mittel) 2 Punkte

Fibonaccizahlen sind ein weiteres Beispiel für Zahlenfolgen, die sehr schnell anwachsen, und ein schönes Beispiel, um den Unterschied zwischen exponentieller, linearer und logarithmischer Laufzeit zu illustrieren.

Die Fibonaccizahlen sind wie folgt definiert:

$fibo(0)=1 \qquad fibo(1)=1 \qquad fibo(i+1)=fibo(i)+fibo(i-1),\ i \geq 1$

Schätzen Sie möglichst genau ab, wie viel Stellen die *n*-te Fibonaccizahl hat, wenn Sie zur Basis 10 bzw. zur Basis 256 dargestellt ist.

Aufgabe 3 Fibonacci-Zahlen II (Votieraufgabe, leicht) 2 Punkte

Schreiben Sie ein rekursives Programm, welches die n-te Fibonaccizahl unter direkter Verwendung obiger Formel berechnet. Verwenden Sie dazu ihr Gzahl-Paket von Aufgabenblatt 11, verwenden Sie für die Anzahl der zur Darstellung nötigen Stellen Ihre Abschätzung aus Aufgabe 2 (falls Sie Aufgabe 2 nicht lösen konnten, verwenden

Sie $1+\left\lfloor \frac{n}{10} \right\rfloor$ für die Anzahl der Bytes).

Geben Sie den Aufwand Ihres Programms in O-Notation an!

Aufgabe 4 Fibonacci-Zahlen III (schriftlich, mittel) 2 Punkte

Wenn man die Fibonaccizahlen von *fibo(0)* ausgehend berechnet, kann man den Rechenaufwand stark verringern. Erweitern Sie ihr Programm um eine iterative und (!) eine rekursive Prozedur zur Berechnung der Fibonaccizahlen, die mit linear vielen Schleifendurchläufen bzw. Prozeduraufrufen auskommen.

Aufgabe 5 Fibonacci-Zahlen IV (Votieraufgabe, leicht) 1 Punkt

Fibonaccizahlen haben folgende Eigenschaft (1<=k<n):
*fibo(n)=fibo(k)*fibo(n-k)+fibo(k-1)*fibo(n-k-1)*
Zeigen Sie diese Eigenschaft durch Induktion über k.

Aufgabe 6 Fibonacci-Zahlen V (schriftlich, schwer) 4 Punkte

Erweitern Sie Ihr Programm um eine weitere Prozedur, die die Fibonaccizahlen unter Ausnutzung der Formel aus Aufgabe 5 berechnet. Schätzen Sie die Laufzeit in O-Notation ab.

Aufgabe 7 Fibonacci-Zahlen VI (schriftlich, leicht) 1 Punkt

Überprüfen Sie Ihre Abschätzungen der Laufzeit durch Tests mit Ihrem Programm (berechnen Sie dazu z.B. die Fibonaccizahlen 10, 20, 30, ..., 100, 200, ..., 1000, ...). Die wievielte Fibonaccizahl können Sie innerhalb von <u>3 Sekunden</u> mit dem Algorithmen aus den Aufgaben 3,4 bzw. 6 berechnen?

Aufgabe 8 Ein String als ADT (Votieraufgabe, mittel) 2½ Punkte

Nun werden wir wieder einen Tick theoretischer und beschäftigen uns mit der Modellierung von abstrakten Datentypen.

Beschreiben Sie den Datentyp String als ADT. Benutzen Sie dabei die Mittel, die in der Vorlesung vorgestellt wurden.

Aufgabe 9 Potenzmenge (Votieraufgabe, mittel) 2½+3 Punkte

Hier sieht man die Anwendung der Modellierung.

a) Spezifizieren Sie die Potenzmenge als einen abstrakten Datentyp nach dem in der Vorlesung vorgestellten Schema.
b) Implementieren Sie ihr Schema, indem Sie ein Programm schreiben, welches Zahlen in eine Menge aufnimmt, bzw. entfernt und nach jedem Vorgang die Potenzmenge ausgibt. Überlegen Sie sich eine sinnvolle Benutzerführung. Die Potenzmenge selbst soll als ADT in ein eigenes Paket ausgelagert werden.