bruk ved søking (vi har ikke lenger en ideell situasjon med direkte oppslag). La oss for anledningen kalle funksjonen finnplass().

En annen og bedre løsning enn de som er skissert ovenfor, er å søke med forskjellig steglengde etter ledig blokk, slik at overløpspostene blir spredd. Dette kan gjøres ved å beregne steglengde ved funksjonen

```
int dobbelthash(long nokkel)
{
   return (1 + nokkel%(antalladresser-1));
}
```

i den samme finn plass-funksjonen. Denne metoden løser problemene med klasing. Vi skal studere hvordan denne algoritmen plasserer postene ovenfor (27, 6, 29) på figuren før vi går over på selve algoritmen.

0	1	2	3	4	5	6	7	8	9	10
11	1	2		15	5	28	7			10
				4	16	39	40			
				26	49	17	18			

Vi har hash(27) = 27%11 = 5 og dobbelthash(27) = 1 + 27%10 = 8. Det er fullt i skuff 5, og vi skal lete etter en ledig plass på en smartere måte enn de påfølgende skuffene i tur og orden. Skuffene 6 og 7 er også fulle, så hvis vi gjorde det så enkelt, ville nøkler hjemmehørende både i skuff 4, 5, 6 og 7 hope seg opp i skuff 8 (vi indekserer skuffene fra 0 og oppover).

Vi søker heller med en steglengde fra dobbelthashfunksjonen. Den første skuffen vi sjekker, blir da
(hash(27)+dobbelthash(27))%11 = (5+8)%11 = 2
og her er det ledig.

Videre: hash(6)=6, og dobbelthash(6)=1+6%10=7. Den første skuffen vi sjekker, blir da 2 (vi teller 4 ut arrayen og 3 fra begynnelsen igjen), og her er det ledig. altså den samme hjemmeadressen som 27.

hash(29)=7 og dobbelthash(29)=10. Det er fullt i skuff 7, og når vi går 10 plasser videre (med modulo) kommer vi til skuff 6. Når vi går ytterligere 10 plasser videre, kommer vi til skuff 5. Slik fortsetter det. Overløpsbanen til 29 blir altså (inkludert hjemmeadressen) 7, 6, 5, 4, 3.

Dobbelthash-funksjonen skal altså ikke brukes direkte på nøkkelen, den finner ikke en ny blokk for overløpsposten direkte. Den brukes derimot for å beregne steglengden i letingen etter en ledig blokk.

8.3.2 Kort sammendrag

- Åpen adressering er veldig dårlig for hashfiler med fyllingsgrad større enn 0.75, spesielt for søk som resulterer i at man ikke finner posten man søker etter.