

# Kombinatorik 1MA020: Information om kursen

Vilhelm Agdur<sup>1</sup>

15 januari 2024

<sup>1</sup> vilhelm.agdur@math.uu.se

Den här filen innehåller – förhoppningsvis – all praktisk information om kursen som du kan tänkas behöva.<sup>2</sup> Hur bonuspoängen fungerar, innehåll i föreläsningarna, et cetera.

<sup>2</sup> Om du saknar någon information här, säg till så lägger jag till den.

Den listade kurslitteraturen för denna kurs är våra föreläsningsanteckningar. Vill man ha en annan behandling av samma material finns ett par förslag på gratis tillgängliga böcker på kurshemsidan, inklusive den bok som tidigare var kurslitteratur för kursen.

## Planering av föreläsningarna

Vi skall totalt ha femton schemalagda tillfällen, varav tolv kommer vara föreläsningar, och tre kommer vara övningstillfällen. Om inget oförutsett inträffar så stämmer tiderna och datumen angedda i tabellen i detta dokumentet, men förlita er hellre på TimeEdit.

Datum	Tid	Planerat innehåll
Ons 22 Jan	15:15	F1: Permutationer och kombinationer, och kombinatoriska bevis
Tor 23 Jan	10:15	F2: Binomialsatsen, kompositioner, multinomialsatsen, och lådprincipen
Mån 27 Jan	13:15	F3: Inklusion-exklusion, derangemang, surjektioner, och mängdpartitioner
Tor 30 Jan	08:15	Övning
Ons 5 Feb	13:15	F4: Sammanfattning av alla räkneproblem, samt cykler
Tor 6 Feb	08:15	F5: Genererande funktioner
Mån 10 Feb	08:15	F6: Fortsättning på genererande funktioner
Tor 13 Feb	13:15	F7: Heltalspartitioner, Dyck-stigar och Catalanantal
Ons 19 Feb	15:15	Övning
Tor 20 Feb	13:15	F8: Grafer och träd
Tis 25 Feb	15:15	F9: Diskret sannolikhets teori, introduktion
Ons 26 Feb	13:15	Övning
Mån 3 Mar	10:15	F10: Slumpvariabler
Ons 5 Mar	08:15	F11: Probabilistiska metoden (Också tre dagar innan deadline för att registrera sig för tentan)
Mån 10 Mars	13:15	Redovisning av inlämningsuppgifter, repetition av övningar

## Tentan

Ordinarie tentamen för kursen inträffar den 20 mars – kom ihåg att registrera er för den minst tolv dagar i förväg, alltså lördagen den 8 mars, det är inte kul att inte få skriva tentan för att man glömt registrera sig.<sup>3</sup> Det går att få max fyrtio poäng på tentan, och för att få betyget 5a, 4a, 3a behövs respektive 32/25/18 poäng.

Omtentor i kursen går i juni och i augusti.

<sup>3</sup> Det hände mig mer än en gång under min kandidat och master...

## Bonuspoäng till tentan

Det finns tre olika sätt att få bonuspoäng till tentan, och du kan maximalt få sju bonuspoäng.<sup>4</sup>

1. Att rätta substantiella fel i föreläsningssanteckningarna, lägga till fler figurer till dem, eller i övrigt förbättra dem.
2. Att lösa övningarna eller göra programmeringsuppgiften från ett av övningstillfällena.
3. Att skriva en sammanfattning av en forskningsartikel inom kombinatorik.

<sup>4</sup> Ni kan alltså kombinera olika sätt att få poäng för att nå upp till totalt högst sju – om ni gör saker som summerar till mer än sju bonuspoäng får ni ändå bara de sju.

Den första av dessa, rättningar av anteckningarna, görs individuellt, och du kan få upp till tre poäng för det – ett per gång du gör en sådan korrigering eller förbättring. Om du lägger till en till figur skall den självklart faktiskt bidra till att göra materialet lättare att förstå – en bild bara för bildens skull räcker inte. Med att "i övrigt förbättra dem" menar jag att förtydliga resonmenangen eller ge bättre förklaringar av materialet. Om du hittar en annan källa än föreläsningssanteckningarna som förklarar samma koncept på ett sätt som du tycker är tydligare kan du alltså få bonuspoäng för att skriva in denna förklaring i föreläsningssanteckningarna.<sup>5</sup>

Till varje av våra tre övningstillfällen kommer vi att ha en samling uppgifter att lösa, samt ett förslag på ett smärre programmeringsprojekt relaterat till ämnet.<sup>6</sup> Ni kan få upp till fyra bonuspoäng genom att skriva lösningar på uppgifterna, och/eller upp till sex bonuspoäng på att lösa programmeringsuppgiften – programmeringen kräver både kod och en förklarande rapport om era resultat.

Slutligen kan ni få upp till sju bonuspoäng till tentan genom att välja en forskningsartikel inom kombinatorik eller en tillämpning av kombinatorik, läsa den, och skriva en kort<sup>7</sup> rapport om vad det är för problem som studeras i artikeln, vad de får för resultat, och en skiss av vilken metod de använder. Jag kommer att lägga upp några olika länkar till var man kan hitta sådana artiklar, men jag kommer också

<sup>5</sup> Självklart kan bara en person få poäng per sak som fixas – om en person hittar ett fel och berättar för sina fem kompisar om det felet, och alla lämnar in en fix av det, så får bara den första som lämnar in det poäng för det.

<sup>6</sup> Eftersom ämnet för denna kurs är kombinatorik, inte programmering, är det okej att lösa denna genom att be t.ex. ChatGPT att skriva de olika funktionerna ni behöver åt er. Detta är så klart inte lika bra för att faktiskt lära sig programmering, och ni måste skriva att ni gjort det i er rapport om ni gör så.

<sup>7</sup> Förslagsvis runt fyra-fem sidor.

gärna med förslag på andra artiklar om ni har något särskilt ämne ni är intresserade av eller bara tycker det är svårt att välja.

De två uppgifterna som görs i grupp skall göras i grupper av inte mer än fem personer, och helst inte färre än fyra.<sup>8</sup> Dessa skall sedan presenteras av gruppen muntligen på ett övningstillfälle, då ni också får feedback på ert arbete.<sup>9</sup> Ni behöver inte ha en powerpoint eller dylikt, men ni ska kunna berätta vad ni har gjort och svara på frågor om det.

### Deadlines

1. Förbättringar av föreläsningssanteckningarna är möjliga ända fram till måndagen den tionde mars.
2. För uppgifterna eller programmeringsuppgiften är deadline två veckor efter övningstillfället faktiskt hölls, förutom om man väljer det sista övningstillfället, i vilket fall deadline är onsdagen den femte mars.<sup>10</sup>
3. För forskningsartikel-rapporten är deadline onsdagen den femte mars.

Förlängningar av deadline är möjliga för gruppuppgiften om ni har ett gott skäl att be om det, och ber om det via mail *innan deadline*, så länge ni inte vill ha tid som löper till efter tentaperioden.

### Indelning i grupper

Ni får själva bilda grupper för att göra uppgiften, och sedan meddela mig om gruppens medlemmar och vilken uppgift ni avser göra, så skapar jag en sådan grupp i Studium. Att flytta sig mellan grupper är möjligt fram till två veckor innan deadline för uppgiften – maila mig om ni av någon anledning vill byta grupp.

### Format för inlämningar

1. Förändringar av föreläsningssanteckningarna görs direkt i  $\text{\LaTeX}$ -källan för anteckningarna, genom en pull request mot det GitHub-repository som de ligger i.<sup>11</sup>
2. Även uppgiftslösningar och forskningsartikel-rapporten skall vara skrivna i  $\text{\LaTeX}$ , och rapport-biten av programmeringen likaså. Dessa lämnas in i Studium, med *både*  $\text{\LaTeX}$ -källan och en renderad pdf-fil.

<sup>8</sup> Om antalet studenter som vill göra uppgiften inte är en jämn multipel av fem får vi så klart göra undantag – men om det är två grupper av två studenter så *kommer* jag att slå ihop dem.

<sup>9</sup> Att delta i denna presentation är *obligatoriskt* för att få godkänt individuellt – om du inte kan delta när din grupp presenterar måste du informera mig om detta via mail innan den gör det, och vi arrangerar då en annan tid för dig att göra det.

<sup>10</sup> Detta är för att övningstillfället faller så sent i kursen – er uppgift måste presenteras muntligen redan den tionde mars, och jag behöver tid att faktiskt läsa igenom ert arbete innan ni presenterar det.

<sup>11</sup> Det finns här: <https://github.com/vagdur/Kombinatorik-IMA020>

Om du behöver någon hjälp med att lista ut det tekniska med att få detta att fungera får du gärna be om det under ett övningstillfälle eller i pausen på en föreläsning.

3. Programmeringen får göras i valfritt programspråk, så länge det är möjligt för mig att köra er kod på min laptop och verifiera att den gör vad ni påstår.<sup>12</sup>

### *Användning av AI*

Som tidigare sagt är det tillåtet att använda AI för att skriva sin kod på programmeringsuppgiften – om man inte vill lägga tiden på att lära sig att skriva koden själv är det ett bra verktyg, och man måste fortfarande förstå problemet för att kunna korrigera koden.

För den övriga delen av kursen och inlämningsuppgifterna får ni lov att använda AI för delar, men om ni försöker att bara mata in texten på uppgiften och sedan lämna in vad den matar ut får ni inte godkänt på det. Det kommer dessutom att skada er när ni skall försöka klara tentan – och AI:n är inte riktigt så smart att det hade funkat oavsett.<sup>13</sup> Kort och gott, var smarta med hur ni använder det – lämpa inte bara över allt ert eget tänkande på maskinen, för det kommer inte att fungera.

<sup>12</sup> I praktiken betyder väl detta mest bara att ni inte kan använda Mathematica eller Matlab, eftersom de är proprietära system.

Om ni är fundersamma på om ert projekt har blivit för konstigt för min dator att kunna köra det, fråga i pausen på en föreläsning.

<sup>13</sup> I mina ovetenskapliga test verkar den kunna hitta en korrekt lösning ibland, men också ibland i ett lika korrekt språkbruk skriva nonsens, eller saker som är subilt fel på ett sätt som nog är svårt för dig som student att upptäcka.