

# Kombinatorik 1MA020: Information om kursen

Vilhelm Agdur<sup>1</sup>

15 januari 2023

<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 inlämningsuppgiften fungerar, innehåll i föreläsningarna och rekommenderade uppgifter, et cetera.

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

Kurslitteratur för denna kurs är *Applied Combinatorics*<sup>3</sup> – men eftersom boken är open source så kommer vi också snart ha en version av boken speciellt anpassad för denna kursen, med alla kapitel vi inte använder bortklippta, för den som vill skriva ut boken eller bara ha en mer hanterlig pdf.<sup>4</sup>

<sup>3</sup> William T. Trotter Mitchel T. Keller. *Applied Combinatorics*. Createspace Independent Publishing Platform, 2017 edition edition, 2017. ISBN 9781973702719. URL <https://rellek.net/book/app-comb.html>

<sup>4</sup> Denna kommer när jag har haft tid att skapa den – förhoppningsvis under de två första veckorna av kursen.

## Planering av föreläsningarna

Vi skall totalt ha femton schemalagda tillfällen, varav elva kommer vara föreläsningar. Tre kommer vara antingen övningstillfällen, tid att komma ikapp planeringen, eller fördjupning i tidigare ämnen, beroende på hur det går i kursen med att hålla tiden och på era intressen. Det allra sista tillfället kommer vara repetition inför tentan. De tider och datum som anges i tabellen nedan kan så klart komma att ändras, men när detta skrivs bör de vara korrekta.

Innehållet i de senare föreläsningarna kan fortfarande komma att flyttas runt eller ändras.

Tillfälle #	Datum & Tid	Planerat innehåll
1	Tis 17 Jan, 15:15	F1: Additions- och multiplikationsregeln, strängar, permutationer och kombinationer
2	Ons 18 Jan, 10:15	F2: Kombinatoriska bevis, binomialsatsen
3	Tis 24 Jan, 13:15	F3: Multinomialkoefficienter, distributioner, och gitterstigar
4	Tor 26 Jan, 08:15	Övning/komma ikapp
5	Tis 31 Jan, 13:15	F4: Induktion samt inklusion-exklusion
6	Ons 2 Feb, 08:15	F5: Fortsättning på inklusion-exklusion
7	Tis 7 Feb, 10:15	F6: Lådprincipen
8	Tor 9 Feb, 13:15	Övning/komma ikapp
9	Tis 14 Feb, 15:15	F7: Genererande funktioner
10	Tor 16 Feb, 10:15	F8: Rekurrensrelationer
11	Tis 21 Feb, 13:15	F9: Fler exempel
12	Ons 22 Feb, 10:15	Övning/komma ikapp
13	Tis 28 Feb, 08:15	F10: Diskret sannolikhetssteori
14	Tor 2 Mar, 10:15	F11: Diskret sannolikhetssteori, fortsättning <sup>5</sup>
15	Mån 6 Mar, 08:15	Repetition

<sup>5</sup> Detta är också näst sista dagen att registrera sig för tentan – glöm inte!

## Tentan

Ordinarie tentamen för kursen inträffar den 15 mars – kom ihåg att registrera er för den minst tolv dagar i förväg, alltså den 3 mars, det är inget kul att inte få skriva tentan för att man glömt registrera sig.<sup>6</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 den 12e juni och i augusti.

<sup>6</sup> Det hände mig mer än en gång under min kandidat och master...

## Inlämningsuppgiften

Vi har en frivillig inlämningsuppgift i kursen, som kan ge upp till sex bonuspoäng på tentan.<sup>7</sup> Uppgiften består av två delar, dels att läsa igenom föreläsningssanteckningarna från en föreläsning och fylla i mer detaljer och rätta eventuella fel, baserat på vad som faktiskt sades på föreläsningen och på boken, och dels att skriva lösningar till uppgifterna i slutet av föreläsningssanteckningarna. Tanken är att ni, när kursen närmar sig sitt slut, skall ha fullständiga anteckningar från varje föreläsning, och lösningar till uppgifterna, så att ni kan studera inför tentan med hjälp av dessa.

<sup>7</sup> Full pott på denna kan alltså lyfta dig nästan ett helt betyg!

Denna uppgift skall göras i grupp – och eftersom det är elva föreläsningar blir det elva grupper, vilket preliminärt bör motsvara grupper om fyra eller fem personer.<sup>8</sup> Indelningen i grupper och tilldelningen av grupper till föreläsningar gör vi redan första tillfället.

<sup>8</sup> Eftersom uppgiften skall vara datorskriven, och lämnas in digitalt, kan det vara bra om varje grupp försöker se till att ha åtminstone en medlem som har sett  $\LaTeX$  innan, eller i alla fall känner sig duktig på datorer.

Ni skall också, för att kunna få poäng på inlämningsuppgiften, göra en kamratrespons på en annan grupp – eftersom målet är att anteckningarna skall bli till en bättre resurs för er är det så klart ni själva som är bäst lämpade att bedöma detta.

Deadline för uppgiften är två veckor efter att föreläsningen hölls. Kamratresponsen vara inne inom en vecka efter att gruppen du skall respondera på lämnat in. Är man sen med en av dessa deadlines kan man max få fyra poäng på uppgiften, är man sen med bägge kan man max få tre poäng på uppgiften. Absolut deadline, efter vilken man inte kan få några poäng alls till tentan, är en vecka efter tentan. Kom ihåg att bonuspoäng endast är tillämpbara på ordinarie tenta – på omtentor får man inte lov att ge bonuspoäng.

## Skriva anteckningar-delen

I denna del av inlämningsuppgiften ombeds ni förbättra de föreläsningssanteckningar som fanns för föreläsningen innan den gavs. Förslagsvis kan ni använda era egna anteckningar från föreläsningen, de gamla anteckningarna från tidigare år, kursboken, eller er favorit-resurs att plugga från. Kom ihåg att om ni använder något

annat än just kursboken och anteckningar, era eller de gamla, måste det antingen vara era egna ord eller källhänvisas till var ni fick det från.<sup>9</sup> Samma regler om plagiat gäller så klart för detta som för allt annat på universitetet.

Denna del kan ge upp till tre poäng, och betygssätts som följer:

- 1 poäng: Fixat skrivfel i matematiken och stavfel i texten. Smärre förbättringar som visar att man har läst och förstått innehållet.
- 2 poäng: Förtydligat bitar i resonemangen som var kortfattade eller ottydliga. Förbättringar som visar att man förstått och förmår resonera om innehållet.
- 3 poäng: Substantiella förbättringar, till exempel lagt till fler exempel i texten, förbättrat bevisen, eller lagt in figurer som illustrerar bevis eller exempel.<sup>10</sup>

### *Skriva lösningar till uppgifter-delen*

I denna del av inlämningsuppgiften ombeds ni, mer traditionellt, att lösa några matteuppgifter. Kom dock ihåg att poängen med det hela är att era kamrater i kursen skall kunna läsa vad ni skrivit och lära sig hur man löser uppgiften av er. Det är alltså inte bara att skriva minsta möjliga mängd ord för att jag skall förstå vad ni menar när jag rättar<sup>11</sup>, utan målet är att din vän i en annan grupp skall kunna läsa vad du skrivit och dra nytta av det.

Denna del kan också ge upp till tre poäng, och betygssätts som följer:

- 1 poäng: Löst åtminstone hälften av uppgifterna, med lösningar som är begripliga för mig som rättare.
- 2 poäng: Löst merparten av uppgifterna, med lösningar som är tydliga och kan förstås också av andra studenter.
- 3 poäng: Löst alla uppgifterna, med tydliga lösningar, och tips för hur man skall tänka om man stöter på en snarlik uppgift inkluderade där det är relevant.

### *Kamratresponsen*

Er kamratrespons består i att ni läser en annan grupps inlämning – både antecknings-biten och uppgifterna – och sedan skriver en kort feedback på den. För delen med anteckningarna skall ni svara på följande frågor:

- Är det gruppen har gjort en förbättring av anteckningarna? Har dem blivit bättre för dig att använda?

<sup>9</sup> Om ni har frågor om eller behöver hjälp med hur man får in källhänvisningar, fråga gärna.

<sup>10</sup> På föreläsningarna ritar vi ju ofta bilder på tavlan – om du är händig med att rita grafiker på datorn kan du knipa tre billiga poäng på att ta dem och rita upp dem och lägga in i föreläsningsanteckningarna.

<sup>11</sup> Vilket ju tyvärr tenderar att vara hur många skriver sina lösningar på en tenta. Att ha fler ord och mer förklaring ger dig aldrig färre poäng på en uppgift...

- Är det något särskilt bra i deras förändringar? Är det något som de hade kunnat göra annorlunda?<sup>12</sup>

För uppgifterna skall ni, för varje uppgift, svara på följande frågor:

- Är lösningen korrekt? Om inte, vad är felet?
- Är lösningen lätt att förstå, eller behöver du anstränga dig för att förstå vad de menar?
- Är lösningen användbar för dig när du skall ta dig an snarlika problem?

### *Hur man jobbar på uppgiften*

Föreläsningsanteckningarna är skrivna i  $\text{\LaTeX}$ , så era förändringar av dem måste också vara skrivna i detta system – det är standard för att typsätta matematik, så förr eller senare behöver man lära sig det om man skall syssla med sådant här. Eftersom ni inte skall göra något avancerat är nog den bästa metoden att lära sig  $\text{\LaTeX}$  att prova sig fram och googla. Eftersom det redan finns ett dokument och ni bara behöver redigera i det slipper ni tänka på det mesta av det som kan vara krångligt som nybörjare i  $\text{\LaTeX}$ .<sup>13</sup>

En sak som dock kommer vara ett litet problem är vilket program ni skriver i – till skillnad från ett Word-dokument, som du bara kan redigera i just Word, finns det många alternativ för  $\text{\LaTeX}$  – vi återkommer till vad som vore rimliga val för er här efter att vi har diskuterat git.

Såsom ni kanske sett ligger föreläsningsanteckningarna på GitHub<sup>14</sup>, och just git är ett utmärkt verktyg för den sortens kollaborativa redigering av textfiler (vilket TeX trots allt är) som vi skall syssla med – både för samarbete mellan er i gruppen och mellan er och mig. Git är dessutom ett annat verktyg som alla som sysslar med något tekniskt eller matematiskt förr eller senare tjänar på att lära sig vad det är. Inlämningen av uppgiften sker alltså genom en Pull Request på GitHub.

Att skriva en detaljerad förklaring av hur detta fungerar vore komplicerat – så eftersom ni bara är elva grupper får ni en väldigt översiktlig förklaring här, och kan sedan komma och be om hjälp med eventuella problem i pausen på eller efter en föreläsning, eller via mail. Git är ett system för att samarbeta kring utvecklingen av textfiler – oftast används det för programkod, men det fungerar utmärkt även för TeX. Till skillnad från till exempel Google Docs så sker inte synkroniseringen av era förändringar kontinuerligt, så att ni alla redigerar i samma dokument – istället måste ni avsiktligt välja att en bit av vad ni har skrivit skall infogas i det gemensamma

<sup>12</sup> Se till att vara konstruktiva i er eventuella kritik! Feedback skall vara nyttig för mottagaren för att kunna förbättra sitt arbete, så att bara säga "det här var dåligt" utan att förklara varför och på vilket sätt det hade kunnat vara bättre är varken produktivt eller snällt.

<sup>13</sup> Googlar man "lära sig latex" får man upp guider för hur man börjar från grunden med att skapa ett dokument – men för er borde det räcka väl att googla "skriva roten ur-tecken i latex", eller vad det nu kan vara. (Förslagsvis dock googla på engelska, det ger nog bättre resultat.)

<sup>14</sup> Ni kanske bara har sett kurshemsidan på <https://vagdur.github.io/Kombinatorik-1MA020/>, men den genereras automatiskt baserat på <https://github.com/vagdur/Kombinatorik-1MA020>, där ni kan se TeX-källan till alla filer i kursen.

dokumentet genom att göra en “commit”. Detta hjälper till att hålla reda på vem som skrivit vad, och ser till att man inte kan råka skriva över varandras bidrag utan att märka det.

Det finns, som jag ser det, tre rimliga sätt för er att arbeta, var och ett med sina för och nackdelar. Oavsett vilket ni väljer är dessa steg gemensamma:

1. Varje person i gruppen skapar sig ett konto på GitHub, om den inte redan har ett
2. En person skapar sig en fork av kursens repository. Detta görs enkelt genom att gå in på <https://github.com/vagdur/Kombinatorik-1MA020>, trycka på knappen som det står “Fork” på i övre högra hörnet, och sedan “Create fork” – ni behöver inte ändra något i den menyn.
3. Den person som skapade er fork kommer nu att ha sin egen kopia av kursens material. Om den döpte sitt GitHub-konto till *pellekarlsson* kommer kopian finnas på <https://github.com/pellekarlsson/Kombinatorik-1MA020>. Nästa steg är att Pelle går dit, klickar sig in i “Settings” och sedan “Collaborators” och lägger till resten av sina gruppmedlemmar.

När dessa steg är gjorda har er grupp sin egen kopia av vår kurs innehåll, och alla i gruppen kan redigera den. Nästa fråga är alltså hur ni redigerar, och här har vi de tre alternativen.

- Det finns en redigerare på själva GitHub, så ni kan helt enkelt öppna filen ni vill redigera på GitHub och klicka på den lilla pennan uppe i övre högra hörnet av rutan filen visas i. När du är klar med att skriva finns det en ruta längst ner med rubriken “Commit changes”, där du skriver en beskrivning av vad du just ändrat, och sedan kan trycka på “Commit changes”. Efter ungefär två minuter kommer den motsvarande pdf-filen på GitHub att ha uppdaterats till den senaste versionen<sup>15</sup>, och du kan se hur dina förändringar ser ut i den kompillerade versionen.

Den här metoden har den stora fördelen att det inte kräver att du installerar något på din dator – allt kan ske på vilken webbläsare som helst – och inte kräver någon större förståelse för hur git fungerar. Nackdelen är så klart att det tar två minuter mellan att du har skrivit något och att du kan se hur det ser ut i den resulterande filen.<sup>16</sup>

- Ni kan kopiera all TeX-koden till ett projekt på Overleaf, redigera där, och sedan copy-pastea tillbaka era förändringar in på GitHub i redigeringsverktyget där. Overleaf är som Google Docs fast för  $\text{\LaTeX}$ , så det har alla för och nackdelar som den sortens redigering har – och att flytta text fram och tillbaka mellan git och Overleaf

<sup>15</sup> Vi uppnår detta genom en GitHub Action som automatiskt kompilerar om TeX-filer vid varje commit, om någon tekniskt sinnad person undrar hur detta är möjligt

<sup>16</sup> Om du vill se live hur en ekvation kommer se ut i den resulterande filen kan jag tipsa om <https://latex.codecogs.com/eqneditor/editor.php> – den innehåller dessutom knappar för att skriva de vanligare uttrycken, vilket kan bespara er en del googling om ni inte lärt er kommandona utantill än.

kan bli en huvudvärk. Men det löser problemet med två minuters väntetid mellan redigering och kompilering. Det här är inte den lösning jag hade valt, men den kanske passar för vissa.

- Ni kan skapa er en lokal kopia av filerna på GitHub genom att göra en clone av repositoryt, och sedan redigera filerna i en TeX-editor på er dator. Själv använder jag GitHub Desktop<sup>17</sup> för att hantera git-biten, och redigerar TeX-filerna i TeXworks eller Visual Studio Code<sup>18</sup>. Denna lösningen kräver, som ni förstår, mer mjukvara på er dator, och lite mer förståelse för hur git fungerar<sup>19</sup>, men när man väl har fått den i ordning slipper man både väntetiden för att se effekten av sina förändringar (eftersom man kan kompilera filen lokalt på sin egna dator) och krånglet med att kopiera fram och tillbaka från Overleaf, som de andra metoderna led av.

Varje person i er grupp kan själv välja vilken av dessa tre metoder den vill använda – det är inte nödvändigt att alla gör på samma sätt. Det är en av fördelarna med just git och TeX, att det finns många sätt att arbeta som alla kan samexistera.

### *Hur man lämnar in uppgiften och gör kamratresponsen*

När ni är redo att lämna in så skapar ni en Pull Request<sup>20</sup>, vilket är ett sätt att be mig att infoga era redigeringar av filerna i dokumenten som alla ser. De som skall ge er kamratrespons kommer då kunna kommentera på denna pull request, och jag kommer kolla på den och poängsätta den. Sedan, möjligen efter att ha fixat till eventuella problem i er inlämning, kommer jag att infoga ert arbete i våra gemensamma filer, så alla studenter kan dra nytta av ert arbete.

Kamratresponsen sker som sagt genom att ni kommenterar på pull requesten gjord av den grupp på vilken ni skall respondera.

### *References*

William T. Trotter Mitchel T. Keller. *Applied Combinatorics*. CreateSpace Independent Publishing Platform, 2017 edition edition, 2017. ISBN 9781973702719. URL <https://rellek.net/book/app-comb.html>.

<sup>17</sup> <https://desktop.github.com/>

<sup>18</sup> TeXworks finns på <https://sourceforge.net/projects/texworks.mirror/> och VS Code på <https://code.visualstudio.com/>. VS Code är antagligen lite överkill om man inte redan har det installerat, men till skillnad från TeXworks har det en del autocomplete och annat man kanske är van vid från Overleaf eller programmering.

<sup>19</sup> Med en lokal kopia räcker det inte att bara trycka commit, man måste också push-a sin commit till versionen på GitHub.

<sup>20</sup> GitHubs dokumentation förklarar hur man gör här: <https://tinyurl.com/howToPullReq>, men den är kanske lite teknisk, så be gärna om hjälp om det blir svårt